Improving Data-sharing and Policy Compliance in a Hybrid Cloud

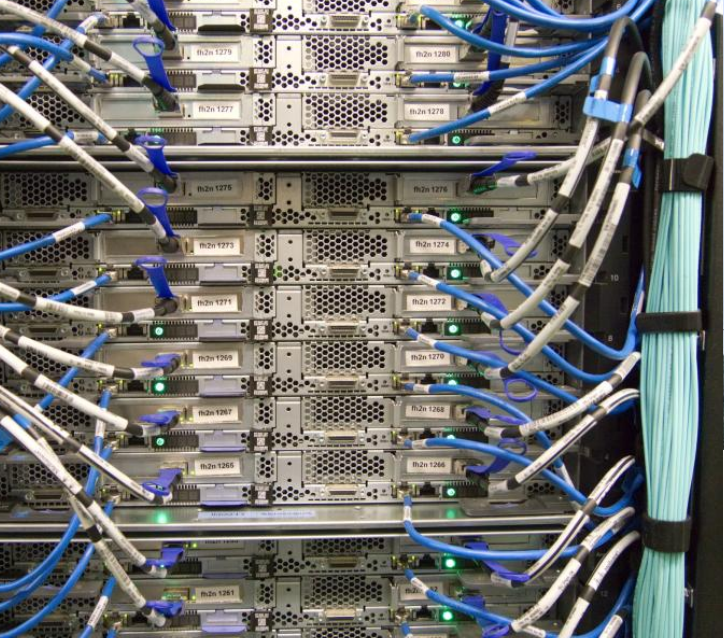
The Case of a Healthcare Provider

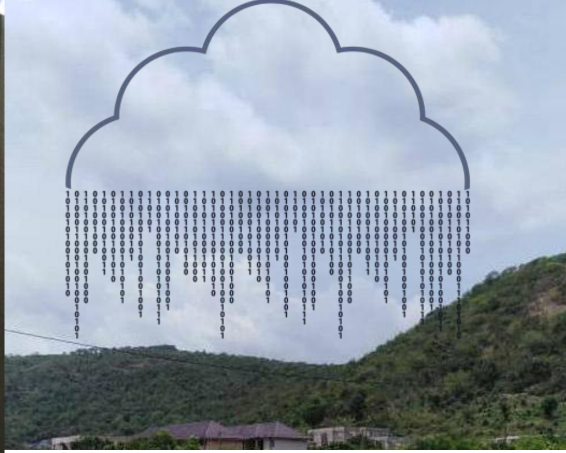
Kwame Azumah, Kenneth

DOI (link to publication from Publisher):
10.54337/aau519583971

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record





```
patientA, /consult, May 31 18:48:14, 10.0.0.1
patientB, /consult, May 31 18:48:15, 10.0.0.1
patientA, /lab,     May 31 18:48:16, 10.0.0.1
patientC, /consult, May 31 18:48:18, 10.0.0.1
patientA, /,        May 31 18:48:19, 10.0.0.1
patientB, /lab,     May 31 18:48:20, 10.0.0.1
```

IMPROVING DATA-SHARING AND POLICY COMPLIANCE IN A HYBRID CLOUD: THE CASE OF A HEALTHCARE PROVIDER

BY

KENNETH KWAME AZUMAH

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY

DENMARK

Improving Data-sharing and Policy Compliance in a Hybrid Cloud: The Case of a Healthcare Provider

Ph.D. Dissertation

Kenneth Kwame Azumah

Aalborg University
CMI - Department of Electronic Systems
Fredrik Bajers Vej 7B
DK-9220 Aalborg
Denmark

Dissertation submitted: September 2022

PhD supervisor: Associate Professor Lene Tolstrup Sørensen
Aalborg University

Assistant PhD supervisor: Associate Professor Sokol Kosta
Aalborg University

PhD committee: Associate Professor Jens Myrup Pedersen (chairman)
Aalborg University, Denmark

Professor Jesus Carretero Pérez
University Carlos III of Madrid, Spain

Associate Professor Daniel Enrique Lucani Roetter
Aarhus University, Denmark

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Electronic Systems

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-828-1

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

This document was typeset using LaTeX. *CloudSim* was used for task scheduling simulations; *MoBuCon tool* was used for process mining and visualisation of constraint violations; *OpenStack Octavia* was used in load-balancing experiments and logging; *CPN Tools* was used to simulate and visualise dynamics of load-balancing in a hybrid cloud and *GNU Octave* was used in generating the plots.

Printed in Denmark by Stibo Complete, 2022

Acknowledgement

"You can make many plans, but the Lord's purpose will prevail." – Prov. 19:21 (NLT)

Finally, at the end of this Ph.D. journey, I have the singular privilege of thanking personalities who have helped in divers ways to bring the work this far. To God be all the glory!

Firstly, my deepest appreciation to the Chair and Members of Assessment Committee who agreed to spend valuable time in evaluating this thesis.

Secondly, a major thanks goes to Professor Knud Erik Skouby, immediate past director of the centre for Communication, Media and Information technologies (CMI) for giving us the opportunity to work at the Center and paving the way for us to learn at many conferences and workshops. A major thanks also goes to Professor Reza Tadayoni, present director of CMI, for playing the initial role of technical supervisor at the formative stages of my Ph.D. journey.

I express a deep and heartfelt gratitude to Professors Lene Tolstrup Sørensen and Sokol Kosta, my inspiring supervisors, who laboured day and night to ensure the rough ideas blossomed into interesting article publications.

Many thanks to Professor Anders Henten, the Ph.D. School liaison in CMI, for advising on the legal fulfilments and also to the entire leadership of the AAU Ph.D. School for the continuous efforts to ease the academic lives of Ph.D. Fellows.

To my co-authors Prof. Paulo R.M. Maciel and Prof. Raffaele Montella, I say thank you for your contributions and for the opportunity to collaborate.

I am also thankful for the support I enjoyed from Staff and Ph.D. Fellows of CMI, with special mention to Anette Bysøe, Idongesit Williams, Alexander Osei Owusu and Joseph Wireku for their divers helps.

A special thank you to my parents, siblings, wife and children for the love, support and understanding over the years on this arduous but exciting journey.

"We must be willing to let go of the life we planned so as to have the life that is waiting for us." – Joseph Campbell

Abstract

Hybrid clouds are composed of two or more individual clouds, most often a mixture of the private and the public cloud. The hybrid cloud therefore enjoys the benefits inherent in each deployment model such as having maximum control over private resources whilst enjoying near-boundless public resources. Operating a hybrid cloud, however, comes with its own challenges as the bringing together of two different cloud computing paradigms introduces more complexity in management, especially in managing security, privacy and compliance to business rules. Included in the oft-cited premise for the increased complexity of the hybrid cloud are the differences in the underlying physical and logical architecture which in turn reduces consistency of security and data processing across the two deployment models. This is one challenge that is a growing concern for hybrid cloud adopters because it thwarts the goal for the adoption in the first place. Proposed solutions have revolved around partitioning data processing applications and specially configuring them to comply with business rules and also developing controls for the underlying infrastructure for processing the data. The issue with this approach is that rapidly evolving business rules and event data culminates in an undesirably high frequency of application re-configurations and programming in order to maintain compliance. This thesis proposes the adoption of a data- and process-aware mechanism in the hybrid cloud to mitigate the challenge of monitoring data processing for adherence to frequently changing constraints for processing data. The proposed process mining framework can be embedded cost-effectively into mechanisms for partitioning applications and routing data in the combined cloud infrastructure. Process mining evaluates the occurrence of information system events for conformance to an expected sequence and timing. By infusing the feedback from process mining into the mechanisms for scheduling workloads, data- and process-awareness can be brought to bear on hybrid cloud placement algorithms responsible for ensuring conformance. The thesis explores process mining-influenced assignation of resources for hybrid clouds with the intention of maximising compliance to the data processing rules of an adopting organisation. The study proposes a framework that is realised and tested in the OpenStack cloud operating system and also in the CloudSim simulation tool, using event logs from the live information system of a selected hospital. To aid in further evaluating the influence of process mining on task scheduling,

the study builds a coloured Petri net (CPN) which simulates process mining-influenced load balance for the hybrid cloud and offers a robust mathematical foundation to support the proposed framework.

Resumé

En hybrid cloud (sky på dansk, men det engelske ord benyttes for en bedre forståelse) består oftest af en kombination af private og offentlige clouds. Hybrid-cloud'en nyder derfor fordelene ved hver implementeringsmodel, såsom at have maksimal kontrol i den private cloud og en meget elastisk ressourcetilgængelighed i den offentlige cloud. At drive en hybrid cloud er dog ikke uden udfordringer, da kombinationen af offentlige og private clouds har tendens til at indføre et ekstra lag af kompleksitet, især i styring af sikkerhed, privatliv og overholdelse af forretningsregler. Blandt hovedårsagerne til den øgede kompleksitet af hybrid cloud'en er forskellene i den underliggende fysiske og logiske arkitektur, som igen reducerer konsistensen af sikkerhed og databehandling på tværs af de to implementeringsmodeller. Dette er en udfordring, og der er en voksende bekymring blandt hybrid cloud-adoptere, fordi det i første omgang afværger formålet for anvendelsen. Foreslåede løsninger har kredset omkring partitionering af databehandlingsapplikationer og specielt konfigureret dem til at overholde forretningsregler og udfører kontrol for den side af hybrid cloud'en, der behandler dataene. Problemet med denne tilgang er, at hurtigt udviklende forretningsregler og hændelsesdata kulminerer i en uønsket høj frekvens af applikationskonfigurationer og programmering for at opretholde overholdelsen af reglerne. Denne afhandling foreslår anvendelsen af en data- og procesbevidst mekanisme i hybrid cloud'en for at afbøde udfordringen med at overvåge databehandling for overholdelse af de ofte skiftende forretningsbegrænsninger. Den foreslåede proces mining ramme kan integreres omkostningseffektivt i mekanismer til partitionering af applikationer og routing af data i hybrid cloud'en. Process mining evaluerer hændelsesdata fra informationssystemets logfiler i overensstemmelse med en forventet sekvens og timing. Ved at tilføre feedback fra process mining's processen til mekanismerne for planlægning af arbejdsbelastninger kan data- og procesbevidsthed bringes til udtryk i hybrid cloud-placeringsalgoritmer, der er ansvarlige for at sikre denne overensstemmelse. Denne afhandling udforsker en procesudviklings påvirket opgaveplanlægning i hybrid cloud med det primære formål, at maksimere overholdelse af et sæt forretningsregler. Afhandlingen foreslår en ramme, der er realiseret og testet i OpenStack cloud-operativsystemet og med anvendelse af CloudSim-simuleringsværktøjet gemmes og overvåges hændelseslogfiler fra et live informationssystem fra et udvalgt hospital. For at hjælpe med yderligere visualisering af indflydelse af process mining på opgaveplanlægning er afhandlingen ligeledes udbygget med en undersøgelse af et farvet Petri-net (CPN), der simulerer påvirkningen af process mining og dens belastningsbalance for hybrid cloud'en, hvilket præsenterer et robust matematisk fundament for understøtning af den foreslåede ramme.

Contents

List of Figures

List of Tables

Thesis Details

Thesis Title:	Improving Data-sharing and Policy Compliance in a Hybrid Cloud: The Case of a Healthcare Provider
Ph.D. Student:	Kenneth Kwame Azumah
Supervisors:	Assoc. Prof. Lene Tolstrup Sørensen, Aalborg University Assoc. Prof. Sokol Kosta, Aalborg University

The main body of this thesis consists of the following papers.

[A] Azumah, Kenneth Kwame, Tadayoni, Reza and Sørensen, Lene Tolstrup, "Hybrid Cloud for Healthcare Data Sharing and Mobile Access: An Architectural Overview," *Nordic and Baltic Journal of Information and Communications Technologies*, vol. 2018, no. 1, pp. 153–176, 2018.

[B] Azumah, Kenneth K., Sørensen, Lene T. and Tadayoni, Reza, "Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis," *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, pp. 1–8, 2018.

[C] Azumah, Kenneth K., Kosta, Sokol and Sørensen, Lene T., "Load Balancing in Hybrid Clouds through Process Mining Monitoring," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11874 LNCS, pp. 148–157, 2019.

[D] Azumah, Kenneth K., Sørensen, Lene T., Montella Raffaele and Kosta Sokol, "Process Mining-constrained Scheduling in the Hybrid Cloud," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 4, pp. e6025, 2021.

[E] Kenneth Kwame Azumah, Paulo Romero Martins Maciel, Lene Tolstrup Sørensen, Sokol Kosta, "Modelling and Simulating a Process Mining-Influenced Load-Balancer for the Hybrid Cloud," *IEEE Transactions in Cloud Computing*, (Accepted for publication, May 2022).

In addition to the main papers, the following articles have also been published.

[1] Azumah, Kenneth Kwame and Kosta, Sokol and Sørensen, Lene Tolstrup, "Scheduling in the Hybrid Cloud Constrained by Process Mining," *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, vol. 2018-Decem, pp. 308–313, 2018.

[2] Azumah, Kenneth Kwame, "Hybrid Cloud Adoption in a Developing Economy: An Architectural Overview," *Handbook on ICT in developing countries: Next Generation ICT Technologies*, vol. 2019, pp. 147–169, 2019.

This thesis has been submitted for assessment in partial fulfilment of the Ph.D. degree. The thesis is based on the submitted or published scientific papers which are listed above. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Preface

This Ph.D. thesis is a summary of papers published with a motivation to build a monitoring framework for hybrid clouds. The papers mainly explore solutions for mitigating the monitoring challenges arising from the combination of the private and public cloud. A running theme among the papers is the application of process mining concepts to facilitate assignation of resources for hybrid clouds. The exploration of these concepts have been done through simulations, employing both programming and visualisation tools with sample data from an actual hospital information system. This approach helps hybrid cloud adopters to minimise the cost of experimenting with various configurations in order to find the ideal setup. The exploration of these approaches are especially important for regions with under-developed Internet infrastructure and resultant sporadic and costly bandwidth. The hybrid cloud enables businesses located in these regions to harness the full benefits of the hybrid cloud by planning the configuration to conform to their business constraints and avoid off-the-shelf and pre-programmed solutions that turn out to be expensive to operate. Apart from maintaining privacy and security for sensitive business data, another importance of monitoring in the hybrid cloud is to ensure that there is regulatory compliance especially in processing client data. Working as an information systems manager in a hospital for many years, the author has first hand experience into potential pitfalls in adopting the hybrid cloud and their impact on the business operating environment. It is hoped that hybrid cloud adopters and practitioners will find the results of this Ph.D. project to be an effective contribution to their cloud adoption, configuration and operation policies.

Kenneth Kwame Azumah
Aalborg University Copenhagen, September 14, 2022

Part I

Extended Summary

Introduction

1 Opening

The Cloud Computing paradigm has been embraced by many organisations including healthcare, academic, financial services and technological startups, as an effective solution to varying computing challenges in the realm of efficient and cost-effective provisioning of resources to users [55, 69]. Efficiency of business activities is one of the main reasons for introducing computing systems in an organisation but as computing resources have cost implications on finite organisational budgets, every resource has to be optimised and managed to support and achieve intended goals respectively. Of the many decisions to make in deploying cloud computing systems, one obvious influencing factor is the suitability of the cloud offering for the business processes and goals. When such a determination is erroneous, it often results in an inefficient adoption in terms of cost, performance or utilisation of computing resources [79, 126]. In many industries, including financial, healthcare and academic, such a determination will need to be made on a periodic basis to evaluate the suitability of the adoption for on-going business activities [69, 99]. Again with an evolving technology trend, such an adoption may also need to be revisited on a regular basis to maximise the benefits.

Hybrid cloud computing has been employed to solve problems that the private or public cloud alone cannot address [126]. The effectiveness of this adoption in terms of cost, performance and goal-attainment is dependent on the composition and the characteristic response of the hybrid cloud architecture [9, 38]. Studies linking business data processing requirements, such as video conferencing and radiography [20, 23], to cloud adoption, are few resulting in poor insight into the impact of choices of cloud computing architecture on an organisation's activities. For example, in embracing the hybrid cloud, the business processes for an insurance company in normal times will be different from that of a healthcare organisation in the time of an epidemic; and the business processes of a large university will be different from that of a small research institute. There is a need for an architecture-goal fit, (between the architectural composition of the hybrid cloud and business activities that will be processed within) that must be measured to determine the optimal composition or architecture that fits a given set of business activities of the deploying organisation [55, 64]. A study into how the choice of cloud computing architecture impacts on an organisation's activities is therefore highly desirable in terms of cost, performance and fitness for purpose.

One viable approach to determining an architecture-goal fit is via process mining, a method for discovering business processes and their extent of conformance to some given business rules [1, 49]. Mining the information system event data provides insight into the process model of the organisation's business activities that utilise the information system. Where this information system is partly or wholly cloud-based, the insight obtained from process mining can be used to determine an ideal composition of a hybrid cloud and suitability of its architecture towards achieving specified business goals.

The study sets out to investigate the impact of hybrid cloud computing on an organisation's business activities by unpacking the strategies for adoption and how process mining can be used to optimise them. The PhD especially focuses on conformance to business rules during task scheduling, an area of application which belongs in the performance enhancement theme of hybrid cloud computing [11]. The motivation for this theme selection is the indirect impact on the overall adoption in terms of cost, privacy and security.

1.1 Impact of Performance on Cost

Operating cost is a constraint imposed on a business and requires that expense operations are subjected to cost constraints towards the attainment of organisational goals [30, 114]. Cloud bursting for data intensive workloads has a profound impact on inter-cloud communication costs [63, 79, 126]. This is especially true for regions with under-developed internet infrastructure [73, 112], where costs can be prohibitively high. The location of the data in the hybrid cloud therefore impacts on the overall performance and operating cost [113]. Mining the event data to determine majority input data location helps to position the bulk of the data on the portion of the cloud infrastructure that does more data processing. The placement of input data is therefore a potential factor in determining conformance to the business rules governing response and overall processing times, just as the business rules pertaining at a period of time will impact on a cost-minimising hybrid cloud architecture.

1.2 Impact of Performance on Privacy and Security

Privacy, as a major hybrid cloud adoption goal, is a business constraint imposed by policy or a regulatory requirement [48, 82] such as in the healthcare and financial industries where personally identifiable information have to be processed within the confines of the governing data protection law [33, 41, 98]. The more cloud bursts take place, the higher the surface area for potential attack on data and applications [8, 37]. Data in transit and at rest have different levels of risk associated especially in a private-to-public inter-cloud communication [8, 14]. The performance of tasks can be optimised to reduce the need for inter-cloud communication and thereby impact on the security and privacy in the hybrid cloud [113]. The security and privacy policy in the hybrid cloud therefore has the potential of impacting on conformance to the governing business rules imposed on data processing per time. Designing and adopting an appropriate hybrid cloud architecture also has the potential of impacting on its security and privacy configuration.

1.3 Brief Overview of Challenges in Hybrid Cloud Adoption

Hybrid clouds, being a compounding of differing deployment models, attain a level of complexity that challenges the seamless deployment of applications and policies across the infrastructure. Investments in cloud infrastructure depend on the demand profile of the adopter, with medium demand opting for the hybrid cloud and high demand going fully public cloud [55]. This section briefly outlines the hybrid cloud adoption challenges typically faced by organisations that want to augment their private data centre capacity with some public cloud resources.

The first major challenge is the migration of selected services from the confines of the private data centre to the public cloud environment which is most often proprietary with varying stacks of subsystems [8, 96]. Governance gets complex as there is a lack of standardisation across these various proprietary cloud stacks.

If the original applications are not cloud native, provisioning can get costly in the new environments. Costs become difficult to handle during the initial adoption [55], either from an all-private or all-public cloud background, that have existing practices to migrate at reasonable cost. The migration can also result in over- or under-provisioning for the public and private cloud infrastructure, respectively [122].

Security is an often cited challenge in hybrid cloud adoption as its application across different deployment models has the potential to create points of weakness that can be exploited as data moves between the clouds [60, 65]. Finally, maintaining compliance gets more challenging and complex as different policies have to applied evenly across different platforms and vendors.

1.4 Problem Statement and Research Aim

The intended adoption goals of the hybrid cloud tend to be hindered by the latter's increased complexity over the purely private or public cloud. The challenge of monitoring the hybrid cloud for compliance to some business rules is a potential threat to the attainment of its adoption goals [4, 43]. Hybrid cloud adoption goals can be classified into thematic areas under which challenges of optimising service provisioning per demand profile can be addressed [11]. The problem of what composition of architecture supports the proffered solutions has seen little research in the opinion of this study. Likewise the impact of the architectural composition on compliance to business rules is not apparent enough to encourage a clear basis for the adoption of a hybrid cloud. The afore-going gaps make it necessary to evaluate hybrid clouds for compliance to business rules and policies towards the achievement of the adoption objectives, which include enhanced performance, cost reduction, privacy and security [11].

Investigating the challenges and the impact of the architectural composition on the adoption goals will provide an understanding of the pertinent factors to consider when operating a hybrid cloud-based information system. Identifying and understanding these factors facilitate the evaluation of the extent of compliance to the provided constraints. The study results provide the knowledge framework to operators of hybrid clouds in order for them to tailor their architectural composition more closely towards the achievement of business objectives and set their policies to overcome the drawbacks inherent in pure and single model cloud deployments.

Adopters of the hybrid cloud who have to deploy an information system across it, face the unique challenge of ensuring the deployment is efficient enough in terms of performance, cost-effectiveness, privacy and security maintenance [11]. Depending on the demand profile of the adopting organisation, the hybrid cloud composition is tailored to meet the adoption goals. In the healthcare industry, for example, where privacy is highly desired, processing excess loads in a public cloud has to maintain the privacy goal [46]. This can be challenging since clinical pathways for treatment tend to vary from patient to patient and also depends on business constraints such as the reigning regulatory directives from a central authority [33, 41, 70]. Though a healthcare organisation can benefit immensely from hybrid cloud adoption, the fluctuating demand profile on account of the varying treatment pathways makes it challenging to predict future workload characteristics. It is in view of this that a healthcare hybrid cloud architecture is proposed to be adaptive to the frequently changing business constraints. The PhD study analyses the impact of hybrid cloud architecture on the performance of business processes using process-mining techniques. It also evaluates the effect of introducing adaptive mechanisms, into the hybrid cloud architecture, towards maximising compliance to business rules. Using data from an existing hospital information system's logs, a prototype software tool is built to test the concept of using process mining output to improve data distribution in hybrid clouds.

Research Objectives and Expected Outcome

This research aims to investigate and evaluate influencing factors for compliance, of hybrid cloud workload processing, to a set of business constraints. It makes an effort to provide a framework for monitoring hybrid cloud tasks and develop a software prototype tool that validates the framework using real data from a case study.

The PhD project attempts to achieve the following objectives:

(i) Identify organisational processes and constraints that affect the distribution of data in a hybrid cloud-based hospital information system.

(ii) Evaluate performance-efficient locations in which to store and process data in a hybrid cloud-based hospital information system.

(iii) Build a prototype software tool to test the concept of improving data distribution in a hybrid cloud-based hospital information system via process mining.

A main expectation is the integration of the proof-of-concept tool for improving hybrid cloud monitoring via process mining and for validating the proposed framework through experiments that employ data from the case study. Specific expected outcomes are:

- Outline of steps in collecting and transforming event data of the targeted hospital information system.
- Documentation of the proposed framework that models the process mining influenced distribution of data in the hybrid cloud.
- Creation and demonstration of a prototype software tool to aid in the placement of healthcare data in a hybrid cloud, based on information system log feedback.

Research Question

How does a healthcare organisation's hybrid cloud adoption strategy affect the conformance of data processing operations to its business rules? In other words, how does conformance to business rules affect the adoption of a hybrid cloud architecture?

RQ1: What architectural design of the hybrid cloud facilitate the monitoring of compliance to a given set of business rules?

RQ2: What kind of data processing operations of a hybrid cloud impact on its performance and compliance to business rules?

RQ3: Which locations of the hybrid cloud are performance-efficient for a selected hybrid cloud architectural composition and prevailing business constraints?

The study lends its significance from the theoretical treatment and approach to improving the distribution of healthcare data in a hybrid cloud situation, over a limited bandwidth. It encourages the adoption of process-aware information systems to aid in determining suitability of hybrid cloud adoption for an organisation's data processing.

1.5 Research Significance

The research endeavours to contribute original ideas in the area of hybrid cloud adoption by evaluating the thematic strategies of minimising cost, enhancing performance and maintaining privacy and security, with on-demand provisioning of computing resources. The research, by designing and building a prototype monitoring tool, contributes to inexpensive decision-making to determine the appropriate composition of the hybrid cloud architecture that achieves service level requirements and at the same time fulfils conformance to data processing rules. The tool helps bring insight into the relationship between the hybrid cloud architectural configuration and data processing performance with respect to prevailing business constraints. It brings specific knowledge into one of the methods of monitoring the hybrid cloud via the application of process mining and event log transformation.

The case study employed contributes to the knowledge of healthcare data processing in the hybrid cloud where there is a challenge of balancing regulatory compliance with minimising data processing costs. As healthcare gets increasingly digital in nature, the volume and speed of data generated also has to be carefully and cost-effectively processed to maintain security and privacy. The regional location of the case study brings special insight into the quality of service characteristics of the hybrid cloud architecture under inadequate or sporadic Internet connectivity.

Load-balancing, as an essential mechanism for exercising control over hybrid cloud data processing, is explored in detail, bringing data-awareness to the mechanism via process mining. The study contributes specifically in tool-chaining middle-ware to facilitate hybrid cloud data processing that respects an organisation's business constraints. The open source OpenStack Octavia project is utilised in this regard, bringing a significant evaluative contribution, into the experimental business rule compliance research arena, for this cloud operating system. This study also proposes a tool that links the visual design of business constraints to the architectural composition of the hybrid cloud to enable adopters more easily ascertain the level of data processing compliance. The choice and construction of the prototype tool contributes to further experimental research, by providing visualisation feedback for data processing performance and compliance in the hybrid cloud.

Finally the study contributes to the body of knowledge on hybrid cloud monitoring by modelling and simulating task scheduling using the coloured Petri net construct. The mathematical foundations established for the model provides a framework that can be built upon in composing various architectures and measuring the potential impact of business constraints on data processing performance. The framework therefore benefits both adopters and researchers of the hybrid cloud through the concise and robust mathematical reconstruction of hybrid cloud scenarios to facilitate adoption decision-making and academic experiments respectively. The concept of using process mining to monitor the performance of a hybrid cloud is also evaluated in detail to lay the foundation for further research both in industry and in academia.

1.6 Research Limitations

(i) Though the implementation of artefacts is based on an established methodology in design science and information systems research, the data for testing the artefacts is not as diversified as required to generalise across industries.

(ii) The data was obtained from a section of the system logs collated over a period of 3 months. There is a likelihood that the hospital clinical activities were skewed towards a set of ailments prevalent within the period of data collection.

(iii) A portion of the study employed the CloudSim simulator developed specially to create (hybrid) cloud setups for simulation and experimentation. It is difficult to (re-)create real-world stochastic Internet conditions in the simulator for generalisation.

(iv) The time-consuming nature of setting up a live OpenStack cloud with process mining integration makes the experimental setup impractical for repeated tear-down and reassembly, for academic purposes. The magnitude of time involved limited further exploratory experimentation on the OpenStack platform.

1.7 Outline of Thesis

In Chapter 1, the thesis introduces the hybrid cloud and gives a brief overview of the challenges brought on by the extra layer of complexity with respect to the effects of performance on cost, privacy and security. This chapter formulates the aim and objectives, translates them into questions and espouses their significance to industry and academia. It finally provides the limitations of the study in terms of scope, methodology choice, resource availability and generalisation of study results.

In Chapter 2, the thesis presents a deeper background into the hybrid cloud computing paradigm, the state-of-the-art and an introduction into the architectural setup and business operations of the selected hospital case study. The chapter presents a review of literature show-casing related works on task scheduling and monitoring in the hybrid cloud.

Chapter 3 analyses the choice of methodology with respect to the research philosophy applied to the study. It presents the various key methods and relates them to the objectives and expected outcomes of the research. It finally lays out the stages in the development of the artefact resulting from the research, providing the techniques and approach for acquiring and processing the data from the case study.

Chapter 4 presents highlights of the systems analysis and design stage for artefact development. The chapter introduces the implementations of the proposed solution, employing OpenStack Octavia, CloudSim Plus and the Coloured Petri Net. Each implementation is detailed out in the published papers, the summaries of which are also presented in this chapter.

Chapter 5 synthesises the individual published article results by discussing the implementation and the findings of the experiments. It also discusses the fulfilment of the study objectives and expected outcomes, and analyses the potential implications for hybrid cloud adoption and management.

In the conclusion, Chapter 6 summarises reflections from the overall PhD study, the limitations of the study and the conclusions contributed from each of the enclosed papers.

Background and Related Work

2 Background

This section introduces details of the hybrid cloud from a number of perspectives including service and deployment models, motivation for adoption, architectural composition, operating economics, adoption strategies and a case study on present-day challenges experienced in the adoption. The section ends with the scope and context of the study.

2.1 Cloud Computing Definitions

One of the earliest and often referenced definitions espouses as important, five primary features of the Cloud Computing paradigm. They are:

(1) "on-demand self-service" [80] where the consumer can provision the service for themselves anytime they want,

(2) "broad network access" [80] where the standard computer network mechanisms are used to deliver the service to the consumer,

(3) "resource pooling" [80] where the provider gives a service that appears as a blackbox to the end user,

(4) "rapid elasticity" [80] where the service can be provisioned rapidly at scale, and

(5) "measured service" [80] where the service can be metered and billed accordingly.

The ISO 17788[1] adds a sixth characteristic:

(6) "multi-tenancy" which comprises the provisioning of separate server instances on a shared physical hardware.

[1] https://www.iso.org/standard/60544.html

Service Models

According to NIST, the cloud resources served to the customer belong to three categories known as *service models* [80]: *Infrastructure-as-a-Service (IaaS)*, *Platform-as-a-Service (PaaS)* and *Software-as-a-Service (SaaS)*.

IaaS is a provisioning that enables consumers to specify physical computing components like processing, networking and storage elements, and also the operating system for managing them [38, 103]. It enables the consumer to setup IT infrastructure virtually and scale-up on demand, thereby minimising high initial setup costs. IaaS provider products include Amazon EC2, Azure Virtual Machines and Alibaba Elastic Compute Service.

PaaS is the service level where the consumer is given room to deploy their own applications within a programming environment [93, 103]. In the PaaS model, the IaaS offering is augmented with an operating system and middleware to facilitate the programming and deployment of applications. The target group for this service model are developers who need a platform that is ready to host their code and data as well as management tools for integrating applications developed. Major provider products of this service model is Amazon Beanstalk and a spectrum of Microsoft Azure products for specialist domains such as artificial intelligence and machine learning.

SaaS is the service level where the consumer accesses functionality in the form of software applications already fixed in features and specifications [93, 110]. SaaS provide services for the entire spectrum of computing resources where the user only makes use of the software application and does not need to bother about the underlying infrastructure. SaaS providers include Microsoft Office 365 and Google Apps with products aimed at end users of applications who may not necessarily have technical backgrounds.

Thus a consumer may want to have control of resources close to the bare hardware in IaaS where they can freely install their operating systems and other software; or need just to install their application software on a provided platform (or operating system) in PaaS with limited control of the platform; or in SaaS, to simply make use of the provided software application such as email.

Deployment Models

How the cloud infrastructure is organised is referred to as the *deployment model* and there are four models: *public cloud*, *private cloud*, *community cloud* and the *hybrid cloud* [52, 80].

(1) The private cloud is the provisioning of computing resources for an unshared individual use by the consumer. The infrastructure is hosted either on the premises of the consumer organisation or by a vendor, offsite. A private cloud delivers the needed control required by the adopter.

(2) The public cloud is the provisioning for shared use of hardware and

software resources with each consumer obtaining an instance of the service.The public cloud is open to all users who want to subscribe to the service on-demand.

(3) The community cloud deployment model can be likened to a private cloud accessible to a group of organisations with shared interests. Most often the organisations are in the same industry or type of business, making cooperation via the community cloud more beneficial.

(4) The hybrid cloud is a provisioning of the separate private cloud deployment and the public cloud deployment combined into one cloud infrastructure. It allows the consumer to have control in the private cloud as well as enjoy on-demand and flexible provisioning of additional computing resources in the public cloud.

2.2 State-of-the-art in Hybrid Cloud Computing

The hybrid cloud, as a cloud computing paradigm, embraces the concept of having multiple cloud deployment types functioning as one cloud space for the purpose of processing information and related tasks. A distinction is made between the "multi-cloud" which consists of clouds from more than one vendor and hybrid cloud as a mixture to two or more deployment models [19, 38]. A vendor can therefore serve up a hybrid cloud to her clients consisting of different deployment models with the hosting infrastructure of the private cloud located either on the premises of the client or of the vendor. It is possible for a hybrid cloud to be defined as a multi-cloud when there are multiple vendors, with each vendor serving up one portion, in the mixture of the cloud deployment models. The hybrid cloud in itself is a cloud deployment model, as defined by Mell & Grance [80], and can be further combined into a multi-cloud in the situation where there is more than one vendor involved. This typical combination is motivated by the need to harness the benefits, and mitigate the weaknesses in each deployment type. In the next sections, the motivation for the adoption of the hybrid cloud is presented, followed by the architectural composition of the typical hybrid cloud. The next sections also present a preliminary economic overview of the private-public cloud mixture and the strategic uses for its adoption. The current challenges are then briefly discussed to conclude the cutting-edge trends in hybrid cloud procurement.

The Case for the Hybrid Cloud

The hybrid cloud is often considered the best mixture of cloud deployments. It has the advantage of providing maximum control over the management of processes in one portion and almost unlimited capacity in another portion of the mixture [11, 44]. The drawbacks to this deployment model are inherent in the complexity resulting from the mixing of two or more different underlying infrastructures. The combination reduces the visibility for data and applications in terms of control of cost, performance, security and resource optimisation and utilisation [128]. Visibility in the cloud means being able to monitor and have control over data processing, hardware infrastructure and associated software. Having visibility therefore facilitates the monitoring for adherence to an organisation's policies on data processing, performance and security requirements. When not properly managed however, a hybrid cloud adoption can lead to an inefficient deployment and a subsequent failure to attain adoption goals.

Motivation for the Hybrid Cloud

Having an easily accessible computing resource in a public cloud is a boon for businesses aiming to reach both connected employees and a wide range of online customers [72]. The benefit of the public cloud is the seemingly elastic availability of computing resources that can be served up on demand without the typical astronomic upfront costs associated with organisational IT resource expansion [8, 21, 68, 79]. However, the public cloud is deployment model which by virtue of its publicly "shared" nature is deemed as less private than an on-premises cloud infrastructure isolated from public networks or the Internet [54]. Where highly sensitive information is to be stored and shared among targeted users, such private and isolated networks are desired. An organisation can implement and maintain its own custom security stack on top of the on-premises private cloud deployment, and have full control of its evolution, if so desired. In a world where information is the new currency, businesses are motivated to deploy private clouds to main control over their sensitive data. Having a mixture of two or more deployment models therefore brings the best of both worlds of near elastic resource availability on one part and high levels of control on another part of the hybrid cloud deployment [106].

For the theoretical underpinnings of this perspective, a survey of literature reveals very little research covering the area of hybrid cloud computing with process mining. Work done on solving the challenges of hybrid cloud storage centre around using machine learning to proactively scale up or down the computing resources automatically in order to maintain a desired performance [9]. Kuo et al. proposed a software-driven hybrid cloud storage architecture to make data highly available on the network [67]. There is however little known documentation on the effects of these solutions on an organisation's information systems and processes. A study by Huang et al. [49] sought to use existing mined information to further supervise and manage running processes within the cloud architecture. They argue that for a running process, the path of an activity in a system log can communicate its level of efficiency and that process mining techniques can enhance the cost-effectiveness of workload scheduling in the cloud. Further related works are provided in Section 2.5 attempting to make a clearer presentation that the research area of cloud computing with processing mining is under-researched especially in industries with complex business processes such as in healthcare.

Architectural Composition of the Hybrid Cloud

One notable feature of the hybrid cloud is the connection between the different underlying network infrastructures. Owing to the fact that the private portion of the hybrid cloud can be located either on the premises of the vendor or the client, the characteristics of connection between the deployment models defines the metrics of the inter-cloud communication. The metrics include the cost, bandwidth and the quality of the communication channel [16, 36]. A hybrid cloud is thus carefully organised architecturally to ensure the adoption goals are supported and that the integrity of the separate deployment models is not compromised [38]. The architectural composition of the hybrid cloud is presented in the context of preserving broad network access, a characteristic of cloud computing, as defined by Mell & Grance [80]. Broad network access implies the use of standard networking protocols in provisioning cloud resources, in metering and in communications between the client and the vendor. For the purpose of analysis, Mazhelis (2012) [79] refers to the hybrid cloud as composed of open and closed subsystems

Hybrid Cloud Adoption Strategies

The deployment model adopted by the organisation speaks to the business objectives that need to be fulfilled. Where the adoption appears to be the only choice in meeting the strategic objectives, steps are taken to optimise the architecture as much as possible. The motivation for hybrid cloud adoption fall into three main thematic strategies [11, 29, 107, 111]: 1) privacy and security, 2) cost optimisation and 3) performance enhancement. In ranking the strategies, performance enhancement was the topmost reason for the hybrid cloud adoption, followed by privacy and security. Cost optimisation was the least sought after strategy when adopting the hybrid cloud cloud. In the long term, however, cost considerations of adopting cloud computing favour the private cloud over the public cloud.

The Economics of the Hybrid Cloud

A primary adoption goal of the hybrid cloud is reducing cost or having a cost-effective deployment model [72]. The cost of running an on-premises infrastructure is mostly fixed irrespective of the level of usage [126]. The public cloud on the other hand has a usage-based or metered cost. Combining private and public cloud in one deployment model affects the cost based on two main differing factors: the computing capacity used and the data communications between the clouds [79, 126]. Communication between the different underlying infrastructure for the hybrid cloud impacts on both the performance and bandwidth costs and therefore is an important factor when considering the cost of adopting a hybrid cloud deployment model. Comparing the costs in a purely private or public cloud deployment model to the hybrid cloud is however far from straight-forward on account of the differences in pricing across cloud providers. For example, if the notion that the cost of public cloud adoption is always cheaper than in-house infrastructure deployment, then public provider A will rent capacity from public provider B and provider B will also rent capacity from provider A. This in reality cannot be true on account of pricing differences, economies of scale, management overhead costs and dynamic quantity-based pricing [126]. A real world example occurred when Instagram moved their operations to the public cloud to save costs since it was cheaper but again had to move their service to a private infrastructure after acquisition by Facebook[2], resulting in further cost-saving[3].

The cost factors categorised by Altmann and Kashef [7] consider electricity, hardware, software, labour, business premises, cloud service and deployment, among which electricity, deployment and cloud service, are the highest cost impact factors. The authors indicated that the services comprised: a private cloud Internet provision, storage elements and data transfers to, from and between clouds. The deployment factor consisted of migration costs for VMs and has a high impact on costs where the hybrid cloud deployment model relies on many VM migrations such in federated hybrid clouds.

In their total cost of ownership (TCO) approach, Martens et al. [75] list cloud selection decision and evaluation; implementation and configuration; support and training, system failure and maintenance as some the other cost types of cloud adoption. From their analysis, the more complex the deployment model the higher the costs of adoption.

2.3 Background of the Selected Case Study

To further apply the conceptual framework and evaluate the software tool prototype, a natural social or organisational environment serves to provide a suitable methodological framework towards contribution to knowledge. In this research, a hospital[4] in Ghana is employed in a case where the outcome of the evaluatory period is documented in the PhD research findings. Permission was granted by the management of the hospital for its hybrid cloud adoption scenario to be used in the case study. In this regard, some background information was provided on the ICT infrastructure and some selected clinical operations.

The hospital is quasi-governmental and has nine branches scattered across the capital city of Ghana and patronised by clients living in the city and the surrounding communities. It consists six satellite clinics which offer primary healthcare and act as feeders into three hospitals offering secondary and tertiary healthcare. The combined daily attendance is over 1000 patients across all the branches of the hospital offering end-to-end healthcare services equivalent to a public regional hospital.

[2] https://www.datacenterknowledge.com/archives/2014/06/27/instagram-migrates-from-amazons-cloud-into-facebook-data-centers

[3] https://www.cheatsheet.com/technology/why-moving-instagram-to-facebooks-data-center-was-worth-it.html/

[4] https://www.thetrusthospital.com

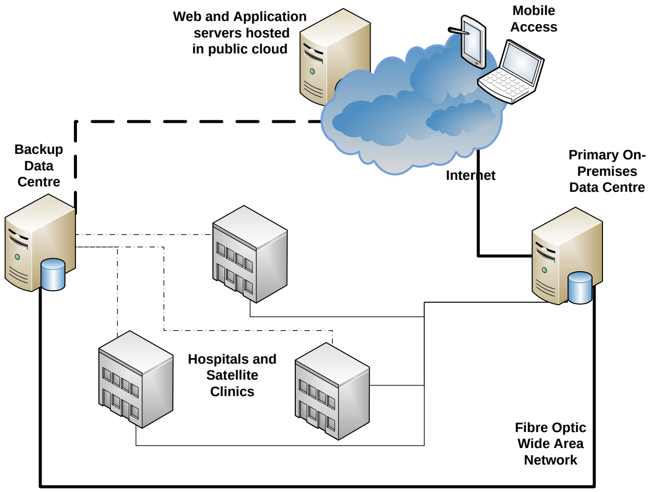


Fig. 1: The hybrid cloud architecture of the selected hospital case study showing how the data centres are connected for both internal and external access by clinicians and clients. The dotted lines are backup connections that are activated during a disaster recovery process.

To support the clinical activities, all the branches are connected via a fibre optic wide area network (WAN) to the hospital's centrally located data centre. The data centre also has a backup infrastructure for disaster recovery and business continuity [102], located in one of the satellite clinics. The hospital not only runs information systems for managing clinical and non-clinical data [92] from its data centre but also rents server space from public cloud providers. Public cloud provisioning enables the hospital to test new applications and also augment its computing resources whenever the need arises. The data centre, again, serves as the primary source of Internet connectivity for all the facilities of the hospital. As a backup to the WAN, each branch has a VPN configuration that tunnels through the Internet and connects to the both the primary and backup on-premises data centres. As a metered backup connection, the VPN is used when the WAN experiences a failure and also employed as the primary remote access connectivity for selected clinicians who connect to the on-premises data centre from their mobile computers, tablets and phones. Most of the data processing is carried out on the on-premises data centre servers however a metered connection is employed allowing data exchange between the data centre and the public cloud for three purposes: (1) when more capacity is needed in the data centre (for instance, to process monthly bills), (2) to collect information from web portals that interact with clients and suppliers in the public cloud, and (3) to push notifications, including medication refill and appointment reminders, to the hospital's mobile apps, downloaded by clients. From the sheer volume of connections, latency is one of the main challenges experienced on the WAN. This necessitates moving some non-critical applications to the public cloud. For example, during monthly bill processing cycles, more public cloud VMs are employed to free up server capacity in the data centre and minimise latency in essential information systems. Figure 1 shows the nature of the connections between the hybrid cloud on-premises components and the rented VM in the public cloud.

The business rules that apply to hospital data processing focus on maintaining the privacy of patients' personally identifiable information [33] and also apply to sensitive results of clinical investigations on patients. Results that are deemed sensitive are programmed as part of the business logic (rules) of the information system. The challenge however lied with the frequency of the reclassification of sensitive data, especially with the on-boarding of new clinical tests and the seasonal upsurge of certain ailments. Per the hospital policy, data classified as sensitive have to be processed in the on-premises data centre to minimise the risk of leakage. Provision is therefore made to increase the capacity of the data centre as the demand for processing sensitive data also increases. Even though the hybrid cloud is expected to absorb the fluctuations in demand for computing resources [13, 66], frequently changing business rules make cloud resource planning challenging with a high potential of violating the rules imposed.

Employing the use case scenario of patient data processing, the PhD study mines the system logs for business rule violations arising from the configuration of the hybrid cloud architecture. The study also aims at helping to decide the optimal configuration per time for some set of business constraints that will be imposed on account of the type and sequence of data being processed.

2.4 Study Scope and Context

Context deals with among other things, the environmental consideration applied to some given information. The context of this study remains in the region of the world where there is limited presence of Cloud/Internet infrastructure and therefore relatively more expensive to adopt Cloud technologies than in regions with more developed infrastructure [76, 111]. The motivation for this choice of context is inherent in challenges posed by hybrid cloud adoption and the complexities in meeting the adoption goals: typically seek to invest in on-premises data centres and supplement with publicly available computing resources when suitable. The circumstances that lead to such adoption generally border on the need to leverage the public cloud to augment locally available resources and meet sporadic computing needs without straining limited and perpetually inadequate IT budgets [76, 84, 127]. This need to embrace the hybrid cloud is more pronounced, in the study's chosen region of poor IT infrastructure, than elsewhere on account of 1) keeping applications and data easily reachable in the event of Internet access curtailment and 2) obtaining a competitive edge by provisioning more computing utility without the upfront investment which will otherwise make such provisioning nearly impossible for many companies operating in such regions.

One may opine that investing long term in IT infrastructure is the way to go for companies that are serious about their future however given the rate of technological advancement and the structure of the regional market [84, 127], it will be financially involving to keep up with relevant technological and security trends, not to mention profiting from such investment entails very high risks [8, 79]. The question then arises of how to provision adequate on-demand computing resources without a heavy upfront investment in infrastructure. Hybrid cloud computing, in the context of a developing economy, wields some viable answers especially when the adoption is suitably done.

The suitability of the hybrid cloud adoption constitutes the scope of this research. The premise for this scope delimitation comes from the backdrop that most business activities are executed towards the achievement of some goal including efficiency, cost reduction and better security or accountability [82]. Where the activity deviates from the intended model for achieving the expected goal, corrective measures must be applied, to put the enterprise back on track to reach the intended goals. This principle applies in the adoption of the hybrid cloud infrastructure where the information systems must be supported to deliver the value for which they were designed [48]. The suitability of the cloud adoption therefore can be determined by how well the business activities are supported in the hybrid cloud-based information system. The scope of the PhD study covers how to measure such suitability of the hybrid cloud to support the dependent business activities. In other words, a measure of how well a hybrid cloud supports a business can be ascertained and a favourable selection made from among various configurations of the hybrid cloud infrastructure [48, 69].

Suitability entails the ability of an entity to satisfy the intentions of adoption. The intentions of cloud adoption are typically economic, performance and security oriented for most businesses [11]. Because of the difficulty in predicting how well a business will do economically, we delimit the scope to the costs of operating the hybrid cloud-based information systems. Performance has indirect impacts on security and cost hence can be measured based on how well it contributes to enhancing security and reducing costs. To move forward in measuring security, the study defines it in the context of the intentions for adopting the hybrid cloud. The intention behind the adoption is to prevent, as much as possible, the unintended processing of classified data in the public cloud. Preventing such unintended leakages of data however has performance trade-offs. The movement of the data and applications across the cloud has cost implications and adequate performance in one cloud deployment model will make inter-cloud communication unnecessary and thereby eliminate or reduce the associated costs. The three-fold adoption benefits of enhancing performance, reducing cost and improving security pertain to the mixing of two different cloud infrastructure under consideration: the private cloud and the public cloud. Measuring the suitability of the hybrid cloud based information system to support a given set of business activities or processes encompasses the inter-relationships between these three thematic areas of 1) performance enhancement, 2) cost-efficiency and 3) achievement of privacy and security. This study sets out to analyse these themes with respect to the case of a selected business entity in the healthcare industry.

2.5 Related Work

This section briefly presents some papers published in the area of task scheduling, and resource monitoring and management in hybrid cloud computing. The studies surveyed employ various mechanisms and techniques for achieving control in the hybrid cloud. The detailed analysis of the surveyed studies are provided in the associated PhD papers and only a summary are presented here to avoid repetition.

Cloud 'Bursting' and Autonomic Provisioning

A main feature of the hybrid cloud is 'bursting' [9, 129, 131], where the excess workload is transferred from one cloud to another, typically private to public, with the aim of satisfying objectives of cost minimisation and meeting time requirements. Mattess [77] proposes a resource allocation policy to aid in dynamic provisioning in order to meet soft-deadlines. The potential loss of security and privacy [88, 130] through cloud bursting can be addressed by discriminating between tasks being transferred across the boundaries of the different deployment models. In [15], the study exploits the location of the workload to achieve time and cost task processing objectives in the hybrid cloud. In [42] the authors introduce a machine learning-based autonomic resource provisioning system. The authors in [62] present algorithms to minimise under/over provisioning in the hybrid cloud. Apart from [130] and [88], the provisioning systems presented are not able to distinguish between sensitive and non-sensitive tasks.

Task Scheduling

Other studies look into the monitoring of cloud resources [4] to help make scheduling and placement decisions. Some studies also look into the optimisation of the decision-making process by employing specific algorithms such as the hybrid bat algorithm [101] and other load balancing techniques [95, 97] to enhance data processing efficiency. A survey of "task scheduling in the hybrid cloud" [27] attempts to classify the techniques and their relative execution costs. A user request scheme in a peer-to-peer architecture is proposed in [124] to optimise the auto-scaling process in the hybrid cloud. Jiang and Sheng [58] also present an algorithm to minimise the cost of hybrid cloud data processing. Bittencourt et al. [18] utilise Directed Acyclic Graphs to schedule tasks with the objectives of mininising time taken as well as cost of execution. In [28] the authors present strategies for exploiting data locality in order to reduce to cost of execution of tasks. Parallelism-awareness is introduced by [121] achieve time goals by utilising all available resources towards the objective. The strategies have however not been applied with respect to data classification. Utilising a heuristic task scheduling algorithm designed with the main objective of maintaining privacy, the authors in [2] achieve the required makespan during data processing in the hybrid cloud.

Load-Balancing and Compliance Monitoring

Architecture monitoring has also been provided by some papers: [71] proposes a stratified approach employing layers of metrics gathered from the architecture; [32] proposes a smart load-balancer using OpenStack; and [68] proposes a flexible storage system employing the hybrid cloud. The studies [74, 81] propose monitoring of the hybrid cloud via the scheduling of tasks to satisfy business rules. In [26], the authors present a framework for monitoring task scheduling by using business process management techniques.

Among studies focusing on simulation tools and frameworks for composing hybrid clouds, [61] presents a tool for simulating task scheduling at low cost. In [6], the author extends the CloudSim simulator and exploits the parallelism offered by cloud computing to meet SLA goals. A CPN modelling and simulation approach is presented in [5]. Gandhi et al. [40] implements an LBaaS named YODA which has at its core high scalability. The authors in [104] maximise cloud resource utilisation with their CPN-based cloud resource allocation simulator. The studies however do not discriminate between the sensitive tasks or workloads.

Methodology

3 Methodology

This PhD study on hybrid cloud computing employs a multi-method approach to deal with the various facets of the paradigm. Both quantitative and qualitative methods provide a framework for answering the research questions and conducting repeatable experiments in composing the hybrid cloud.

The study employs the design science research methodology [17] which involves the specific steps of problem definition, requirements specification, analysis and design, implementation and evaluation. Relating the design science steps to the study, *analysis* corresponds to the examination of event data for requirement specification; *design* refers to cloud components selection; *implementation* relates to the simulations and experiments using the various cloud configurations, and *evaluation* refers to the verification of whether the implementation follows the design specification by observing that design changes affect the output of the implementation.

Within each stage, the PhD project applies a quantitative or qualitative method to systematically synthesise the inputs into the final product of the stage. For instance in the conceptual and planning stage of systems development, a qualitative approach facilitates the problem definition whereas in the testing stage a quantitative method is applied to measure the correlation between the specifications and the implemented features of the artefact.

Each stage is a sub-study of the research that has questions in the form of inputs and has findings consisting of outputs from the stage. The decision to use a quantitative or qualitative method for a stage relies on ontological foundations, such as design science [17], to identity the area of research and the epistemological knowledge emanating from the state-of-the-art [78, 83]. Being a fluid research area, cloud computing tends to branch off into specialised areas at a rapid rate as the paradigm finds more applicative areas that are based on economic, security or performance needs [11]. The following sections analyse the theories and methods employed in the research.

3.1 Research Philosophy and Methodological Choice

The PhD research's viewpoint is positivist in outlook and has employed a mixed method complex design consisting of qualitative analysis, a selection of the tools followed by quantitative statistical measurements extracted from primary data, using the selected tools.

Designing the architecture is a design science research [17, 51] that focuses on the analysis of the performance of the hybrid cloud computing components. The philosophy of science explanation [85] is a deductive explanation where final results are drawn from the initial hypothesis.

Ontologically, design science research involves material artefacts and therefore is apt to have a realistic and creative output. Epistemologically, design science research is often positivist in outlook [17] relying on feedback from actual usage of designed artefacts. Information and computer science research methodologies tend to belong to one or more of the following: experimental, simulation and theoretical [10]. Combining them is often a must, especially in domains where one method alone is insufficient for robustness in validation. The bleeding-edge of cloud computing practice combines algorithms, computer networking and user behaviour analysis to foster useful innovation. The combination of the techniques often need more than one methodology to design and evaluate an engineered solution.

3.2 Systems Development Approach

This section highlights the multi-method approach in this PhD study, attempts to situate its area of research, the strategies and the systematic nature of the approach.

The Information Systems (IS) development approach to research culminates in multiple applicable methodology, where the objectives and methods for each group are interrelated [86]. Multiple methods are adopted for this PhD study because it can be classified as: (1) an applied research where a solution is developed for an identified problem using facets of the IS body of knowledge; [17] (2) an engineering research where the goal is to develop an artefact that is based on the a theoretical design and also has aesthetic appeal [34]; (3) developmental research where there is a an exploratory search for and creation of a better set of steps for arriving at a desired result [47]. IS development research not only encompasses the fore-going but is also driven by the identification of a problem, development of a hypothesis, analysis and expansion of the hypothesis through data collection, and the validation of results via proofs and working artefacts. This means the working system developed at the end of the research can be an argument in support of the hypothesis and evidence of its "proof" [51]. Most often in IS research, artefacts produced are regarded as the contribution to the body of knowledge because they allow further evaluative and developmental research for a measurement of impact and an improvement in the instruction set, respectively [86]. The impact of IS research lives in the improved technology application, user interaction and system function, in the situations where it is difficult to evaluate without a "proof-of-concept" artefact [51, 86].

Situating the Study

Study areas like operating system programming and optimisation algorithms for networking, storage and data processing are in the core of computer science research. However, the integration of systems making use of these concepts fall into the domain of management information systems where practitioners are typically system integrators [86]. The combination of various systems concepts to solve a problem culminates in IS research that has the aim of mostly digitising business activities to improve accessibility, efficiency and performance [117].

A cloud operating system [25, 90] and its associated ecosystem such as users, deployment and service models belong to a multidisciplinary domain of computing, networking and management economics where: the computing aspects concern the processing of data and its performance; the networking aspects concern the routing, accessibility and partitioning of the cloud resources; and the management economics deal with the cost of the cloud adoption in the short and long term with respect to the various deployment models. This PhD study therefore falls in the multidisciplinary area of computer science and management information systems [51, 86]. It builds a framework for system integration toward closer monitoring of business processes and also explores the underlying theories for the resulting subsystems: load-balancer and task scheduler; process mining monitor; and business rule manager.

Strategies in the Study Approach

The strategies used in the study methodology are premised on the multi-methodological approach to IS research [86, 118]. They consist of [86]

(1) the theory building strategy which features the mathematical models and system specifications;

(2) the observation strategy which is concerned with the application on the case of a selected hospital information system log or event data collection;

(3) the experimentation strategy which consists of simulations and testing of the theory's alignment within the context of the observations; and

(4) systems development which is an implementation of the product or artefact from the initial design stage up to the target user evaluation.

In this study, the system development goes up to the prototype stage. This initial product development strategy offers feedback to other strategies with respect to collecting data to validate proof-of-concept. Its ultimate purpose is to provide a vehicle for theory testing and experimentation. The interrelationship between these strategies is depicted in Figure 2 that shows an overall effective approach to IS research.

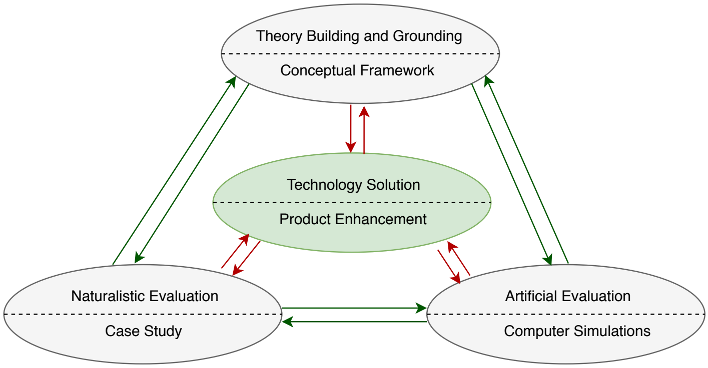


Fig. 2: A Design Science Research Framework [86, 118]

The capstone strategy of systems development consists of five main stages [51, 86, 118]:

(1) conceptual framework construction where a clear definition of the problem is made to guide the development process, build theories or framework within which development is to proceed;

(2) system architecture development where specification and measure-able requirements of the system are defined keeping in mind the desired outcome and functionalities;

(3) system analysis and design, the stage where an analysis of the proposed solution is carried out within the context of the application domain and a model abstracted employing selected components among alternatives;

(4) system building, the fourth stage where a working prototype of the proposed solution is realised in readiness for testing, to confirm feasibility of the solution; and

(5) system experimentation, observation and evaluation where the performance of the resulting system is tested against the functional and technical requirements, to ascertain level of satisfaction under various experimental conditions.

The systems development approach is further expatiated in the next section in relation to the specific implementation of the artefact.

3.3 Implementation Stages for Artefact Realisation

To systematically implement the proposed solution to the problem of the hybrid cloud architectural composition, the study applied the agile model [87] of the systems development life cycle (SDLC) methodology [31, 94]. The SDLC activities can be classified generally into [22]: (1) identification and concept, (2)requirements definition, (3)system design, (4) implementation, and (5) testing. The research follows the five stages and provides details in the following Subsections.

Problem Identification and Concept planning

In this activity the intended users of the system together with the developers put forth ideas that will solve the problem identified. At this stage the users may not have a full grasp of the features possible on one hand and the developers on the other hand may not have full understanding of the domain and are likely to underestimate the problem [22]. The problem is defined with respect to how to overcome the monitoring limitations, on account of architectural heterogeneity of the hybrid cloud, by bringing business policy-awareness to data processing. The outputs of this stage are propositions of various solutions and approaches with their costs and benefits. The inputs of this stage are interviews, a feasibility study and investigations into the existing system, to gain insight into the problem and the desired solution.

Requirements Specification

In defining the requirements both users and developers specify the functional requirements that are modified, as needed, to accommodate the new consensus, built from discussions by the two groups, users on one hand and a developers on the other [22]. This stage in the methodology has foundations in design science where functional systems are conceived, designed and implemented [51]. The specification of the requirements constitutes the research questions that must be answered or investigated [86]. In the cloud computing paradigm, the requirement specification answers questions about the optimal architectural configuration for supporting a desired set of business rules. The inputs to this stage are the results of the feasibility study with the latter also detailing a cost benefit analysis. The inputs provide a scope and boundary that help determine the level of resource investment to apply to the functional and technical design [91].

System Analysis and Design

The analysis and design is the stage of the methodology that systematically lays out the components required to address the problem. In the design activity, the developers specify the interfaces, databases and models that can best be applied to implement the system [22]. In this thesis, the analysis activity reveals the need to focus on inter-cloud communication and a fitting architectural design. When data-intensive processes are executed in the public cloud and have to communicate with a database in the private on-premises cloud, the inter-cloud communication bandwidth usage can: (1) accumulate rapidly if the link is fast, or (2) impact negatively on performance if the link is slow. Such analyses not only fits the methodical approach to studying the problems on cloud computing but generates requirements towards a system (re)design [51]. The requirements now dictate the direction for pursuing the resulting sub-studies and the methods to be applied on each one [86].

Instantaneous artefacts tend to be the products of the design process in the area of computer or information system science research projects especially when such projects are more specific than abstract [86]. This PhD study involves a generative design that proffers a solution to a problem from a standpoint of abductive reasoning [51]. The setup of the experiments generate more insight about an existing system and offers avenues for addressing the specific problem. The study generalises the problem and evaluates each solution avenue against the problem to satisfy explanations for the phenomenon. The creativity aspect of design is employed for both generating the artefact instance and for theorising during the search for the solution. Both the theory and artefact are evaluated and revised in the design cycle.

System Implementation and Prototyping

In the implementation phase, the developer logically follows the facts of the requirements and the design hypotheses and translates them into a working program. It may be necessary for the developer to further seek clarification on component choices so that the best algorithms or models can be selected in the implementation [22]. The feedback also helps to manage the users expectations. In order to prepare to evaluate the design, the system is realised in a social setting or technical environment that has the problems or functional requirements matching the specifications of the project. The conceptual framework that gave birth to the design comes alive through the process of building the system or artefact. The building process serves as an evaluative mechanism for the design stage where alternative solutions can be incorporated into the implementation. At this stage, the contribution to existing knowledge comes from the experiences gathered during the building process, offering more clarity beyond the conceptual framework where the knowledge is more “cloudy” [86]. The building of the system also sets the stage for further experimentation and knowledge gathering helping to refine the concepts and underlying theories.

Testing, Operation and Theory Building

Testing involves comparing the finished product with users’ expectations. The users can also gain knowledge about new possibilities introduced during construction of the artefact [22]. This phase connects the underlying theoretical framework to the end-product of the implementation. The explanatory design theory [17, 118] that is applied in our theory building has two parts: requirements and the solution components which are respectively the starting and ending points in the design research. The PhD research applied a deductive explanation approach towards attaining the conclusions to the study [17]. In contributing to the design science research field, the PhD study is halfway between a specific and general solution where operational principles or architecture become the result of the methods applied [118].

The solution applies abduction as a process to obtaining conclusions from the hypothesis. The PhD's technological solution is a proof-of-concept prototype designed with the intention of gaining insight into the problem environment and designed as a tool to support business artefact decision making adjustments.

3.4 Application of Theory and Key Methods in Data Collection

This section analyses the underlying theories governing the production of the artefact and lays out the key methods employed in data collection. The building of a prototype monitoring system for hybrid cloud data processing, as part of the research end-product, falls in the *design and action* category of information systems development theory (ISDT) [45, 59].

Applying Information Systems Development Theory

ISDT has two main parts [119]: the *design product* and then the *design process*. The design product component comprises [50, 119]: meta-requirements, meta-design, kernel theories and testable design. The components have been employed by [120] in developing a laboratory system.

1. **Meta-Requirements:** The meta-requirements component facilitates problem definition and design goal classification to situate it within the theory. The PhD study considers four classes of inputs to define for the hybrid cloud monitoring system: (a) specification of the cloud technologies and providers, (b) definition of the target users and environment (c) definition of the level of support needed for the application domain with respect to the flexibility of the artefact end-product and expertise of technical support, and (d) specification of the capacity of users to programme and utilise the system across varying business constraints. The meta-requirements component of the theory is equivalent to the conceptual, problem definition and initial analysis stages of the SDLC.

2. **Meta-Design:** Meta-design principles allow for formulating products to fulfil the meta-requirements. This component of the theory focuses on the integration of the various product modules of the system with the aim of meeting the business rule requirements of each domain as well as

flexibility to accommodate varying event logs. Meta-design answers to the analysis and design phase of the SDLC.

3. **Kernel Theories Governing Design Product:** Kernel theories facilitate the incorporation of social and natural science theories that are influencing the artefact design. Given the dynamic nature of healthcare processes, the main governing theories that answer to the design requirements with respect to the hybrid monitoring system are (a) Process Mining theory - to discover changes to existing process models and facilitate conformance checking [1, 3, 115], (b) Business Process Reengineering [100, 116] to facilitate the needed restructuring to take advantage of the hybrid cloud monitoring system, and (c) Task-Technology Fit theory to measure the value of the artefact introduction to the organisation's processes [24, 105]. The kernel theories component relates the linking of requirements definition to the proposed design.

4. **Testable Design Product and Process:** The design product and process have hypotheses that test for fulfilment of requirements. For the artefact development, they are:-

   - The Hybrid Cloud Monitoring System (HCMS) is independent of the providers and the underlying IaaS infrastructure.
   - The HCMS is a cost-effective tool for detecting compliance to varying business rules.
   - The HCMS is a decision support tool in cloud adoption planning and for speedy changes in architectural response to business needs
   - The HCMS contributes to building trust, between technology managers on one hand and users on the other, that business rule compliance can be maintained amidst the adoption of new cloud systems.

Applying Key Methods in the Design Process

The design process component comprises the design methods for developing the artefact, kernel theories on the design process and testable design process hypotheses to validate the results.

1. **Design Methods:** This is a description of the artefact construction process laying out the methods and rational approach. It lays out the procedure employed such that the artefact can generally be reproduced with similar results. The construction of the prototype artefact followed the system development life cycle for software tools comprising the problem definition, requirements specification, system design, coding and testing.

2. **Kernel Theories Governing Design Process** The design process employs the kernel theories engaged in the design product. The process mining theory aids in refining the focus of the artefact by providing more insight into event data.

The meta-design hypothesis is compared with the hypotheses of the design. Comparing hypotheses therefore allows both empirical and interpretative validation of the results.

Employing a case study of a selected hospital

Pertinent to research in process mining is the information system log data that will serve as input for analyses. The case of a selected hospital[5] in Ghana that sees over one thousand patients a day will serve to provide tens of thousands of event data from its hybrid cloud-based information system logs. Its hospital information system is a combination of connected subsystems integrated in a service-oriented architecture. The case study is employed as a method to acquire in-depth analysis of the research area within the hospital. A close look the physical business processes on site helps to facilitate the comparative analyses within and without the hybrid cloud architecture.

Applying Process Mining Theory and Methodology

The primary data in the research is obtained from the information system logs of the selected hospital case study. Process mining theory is [1, 3, 115] applied in extracting the raw data to gain initial insights into the processes in the hospital. The focus was in the outpatient department of the hospital consisting primarily of the processes: consultations; laboratory and radiological investigations; prescriptions and medication dispensing; and review consultations. It is important to pose initial questions in order to help focus the extraction of the data from the various system logs. The questions revolved around the identification of the activities culminating in the afore-listed processes. The thesis adapted the process mining methodology featuring six stages [117]: "planning, extraction, data processing, mining and analysis, evaluation, and process improvement".

1. **Planning** The goal for extracting and analysing the data is established at this stage. The research questions are formulated from the intended objectives and hypotheses culled from the objectives. The objectives help identify the selection of data systems to include in the data extraction and the selection of business procedures. The planning stage is an iterative process of cleaning the event data, determining their relevance and identifying the research questions that can be answered.

2. **Extraction** stage has the goal of extracting event data and the process models from the system logs. The main activities in this phase are (a) Determining the scope with respect to the granularity, period duration, attributes of the data to collect and the plan to correlate the data from various logs. (b) Extracting event data from across information systems

[5] http://www.thetrusthospital.com

into a single collections, and (c) Transferring knowledge of the domain from the experts to the analysts for a better understanding in the view generation. The inputs to this stage are the research questions and the information systems identified. The outputs are the compilation of extracted data and identification of perspectives of the event logs gathered.

3. **Data Processing** is the stage where the goal is to create different views of the log. The inputs are the collection of event data and the output is the event log which is arranged from various perspectives. The activities for this stage involve (a) filtering through the process models (b) aggregating the events hierarchically (c) enriching the logs by computing an attribute based on others and (d) creating views from event class, case, temporal or resource perspectives.

4. **Mining and Analysis** is a phase which takes as input the event logs and outputs findings related to the performance or compliance research question. The activities involve (a) process discovery which returns a process model (b)conformance checking that detects inconsistencies in execution log on the given process model (c) enhancement which extends or improves on an existing process model.

5. **Evaluation** stage takes as input the process models, performance and compliance findings; and outputs improvement ideas and new research questions. The activities at this stage involve (a) diagnosing insights to obtain correct interpretation of results; distinguish unusual patterns; and help refine the research questions (b) verifying and validating by comparing the findings to the original data or system and by comparing the findings to the claims of stakeholders respectively.

6. **Process Improvement and Support** is a stage in the methodology that takes as input improvement ideas and outputs process modifications. The activities involved in this stage are (a) implementing improvements and (b) supporting operations with the new knowledge gained.

Applying Process Mining Tools

Process mining is a research area derived from a combination of "machine learning, data mining and process modelling and analysis" [1], the main goals of which are to obtain information from system logs. "Process Discovery" [1] in process mining focuses on the control-flow perspective and the task of identifying processes [1]. The process discovery task uses algorithms to relate event logs to representative models of the business operation generating the event. One of the first algorithms, the α-algorithm, deduces patterns from the event log such as: if an event Y always follows an occurring event X but not the other way round, then a causal dependency exists for the events X and Y [1]. Such ordering relations are established between the events and a graph generated out of it. The α-algorithm is however susceptible to noise [1], especially to incomplete data in the event logs, and therefore is not practically used in process mining. Some existing algorithms for identifying processes in event logs include the heuristics, inductive and fuzzy miners [1]. Each one has drawbacks to usage but this study applies the Heuristics Miner to discover the processes [35, 123, 125] because it is able to handle event log noise much more efficiently. Event information gleaned from the selected hospital's logs are extracted, cleaned up and fed into ProM 6 [53], a popular data mining research tool. Data cleaning is done via filtering, to reduce noise (low quality data and outliers) and then the heuristic algorithm is applied to unearth the business operating model of the hospital. A section of the log data compiled from the selected organisation is shown in Table 4 of Paper D. An initial processing of the event log utilising the *Visual Inductive Miner* in ProM 6 yields the output shown in Figure 3.

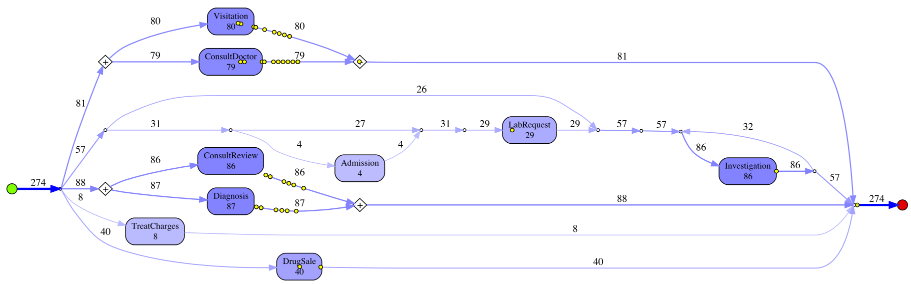


Fig. 3: The output of the initial processing of a section of case study event logs using the ProM 6 Visual Inductive Miner.

Modelling and Simulation of the Proposed Framework

The PhD study employs the Coloured Petri Net (CPN) [57] as a tool for generating a working model of the proposed tool-chain. The construction of the CPN model follows the framework of integrating components where each one ingests data, processes it and outputs information for the next component in the tool-chain. The CPN as a tool has a robust mathematical foundation that allows the concise description of concurrent processes and a complete but compact representation of all states of a system. Coupled with a visualisation of the state-space, the CPN can be used to measure timings of data processing, lengths of queues and distribution of data per configuration. The PhD study constructs a CPN of four integrated components to observe and measure the characteristics of a hybrid cloud under various proportions of sensitive information, determined by a specified set of business rules. The CPN's simulation view takes as input an assemblage of tasks and business constraints, and generates metrics of the overall processing time, task queue lengths, throughput, distribution of tasks and bandwidth costs, for each configuration of a number of VMs for the modelled internal data centre. The construction and fulfilment of the CPN is further examined in Section 4.1.

Development and Validation of Software Prototype

To efficiently gauge the performance of business processes within the hybrid cloud architecture, the log data has to be continuously analysed. ProM 6 [53] as a complex academic tool is not well suited to apply as a continuous analysis software tool mainly because of the numerous human operational steps needed to come out with a meaningful result. The main requirement of the software prototype is to automate as much as possible the steps involved in extracting the process model and measuring the performance of selected business processes. Chain-calling ProM 6 APIs [53] to extract and analyse event log data is expected to achieve the same output as with manually using the ProM 6 tool. To verify and validate the software prototype, a sample is manually analysed in the ProM 6 tool and the same data is analysed using the prototype software. The end result of both manual and the automated analysis should be identical. Simulations will be carried out using test data sets to further validate the software. The advantage of using the software prototype in the analysis should be evident with from the speed of extracting, transforming, loading and analysing event log data.

Implementation Results and Summary of Published Articles

4 Implementation Results

This section details the main artefacts produced from the implementation of the proposed framework. The summary of published articles describing the development and experimentation with these artefacts is also presented.

4.1 Summary of Implemented Prototype Artefacts

The main objective of developing the prototype artefacts include monitoring task processing in the hybrid cloud. The developed systems have the ability to distinguish between tasks for private (sensitive tasks) and public (non-sensitive tasks) cloud processing and checks if tasks for private cloud processing inadvertently wind up being processed in the public cloud. The composition of the hybrid cloud architecture exerts a noteworthy influence over task processing efficiency and therefore the artefacts offer cost-effective solutions to testing out the hybrid cloud adoption under varying loads of sensitive tasks. This testing facilitates the appropriate selection of cloud resources in order to compose a near-optimal hybrid cloud architecture for meeting quality of service requirements. Three perspectives of the hybrid cloud monitoring artefact are presented. They comprise the implementation in OpenStack Octavia, the implementation using CloudSim Plus and the Coloured Petri Net representation. Each implementation targeted a set of hypotheses enumerated under Section 3.4, information system design theory.

OpenStack Octavia Implementation

Octavia, a "load-balancing-as-a-service" (LBaaS) project [108] in OpenStack, shown in Figure 4, has the ability route network packets at the application layer (OSI layer 7) giving an extra advantage over commodity hardware load-balancers which route at the more basic data link and network level (OSI layer 2 and 3). The Octavia project [108] features a layer 7 (L7) database that stores programmable policies for influencing the load-balancer's routing decisions. The L7 rules, comparison types and policy actions are shown in Table 1. The study setup calls the Octavia API to create new rules in response to a process mining logical output for determining conformance to some set of business constraints. The process mining is based on the system log that is transformed into event data. The data flow of the entire tool-chain is depicted in Figure 1 in Paper C and Figure 4 in Paper E where the output of one stage for the process mining serves as an input for the next module. Paper C presents the details of the OpenStack Octavia setup and the section of the event data captured from the log files.

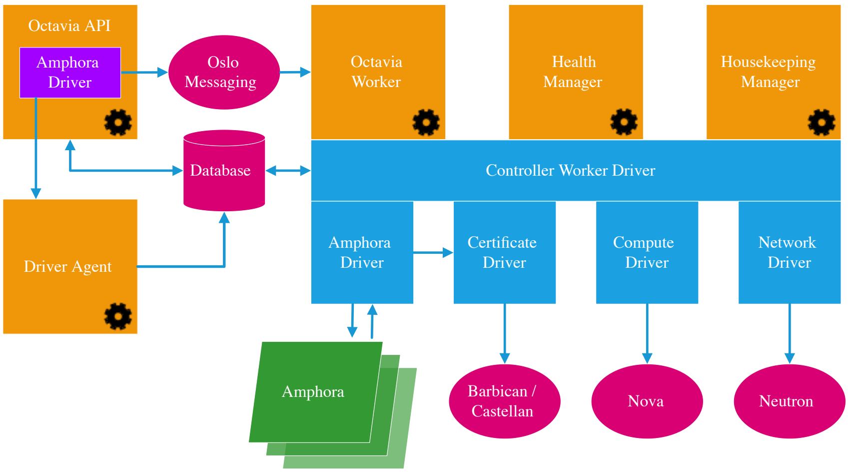


Fig. 4: The architectural layout of the OpenStack Octavia Load-Balancing-as-a-Service project showing the components and their relationship to each other (obtained from the OpenStack Octavia [108] project)

CloudSim Implementation

CloudSim Plus [39] is a Java-based toolkit for simulating the behaviour of entities and for modelling cloud deployments of various configurations at a fraction of the cost of the actual cloud deployment.The framework provided by the CloudSim Plus toolkit facilitates the setup of entities such as data centers, virtual machines and processing elements including multi-core processors, RAM and persistent storage. The toolkit is extensible via programming to create cloud configurations for further exploration. In the setup used for this project,

Table 1: Types of L7 Rules, Comparison Types and Policy Actions [109]

RULE TYPES
HOSTNAME - compare between HTTP/1.1 hostname in request and value PATH - compare path portion of HTTP URI and value FILE TYPE - compare last portion of URI and value (eg. txt, png, etc) HEADER - compare header defined in key and value COOKIE - compare cookie named in key and value
COMPARABILITY
REGEX - matching by regular expressions STARTS WITH - compare starting portion of value ENDS WITH - compare ending portion of value CONTAINS - compare substring of value EQUAL TO - compare equality NOT - inverts comparison
POLICY ACTIONS
REJECT - denied with appropriate code REDIRECT TO URL - request sent an HTTP redirect REDIRECT TO POOL - request forwarded to backend pool

the modules are the `datacenterbroker` class to offer data-awareness to task scheduling. The goal of the configuration is to emulate task processing in the hybrid cloud and observe the dynamics of sensitive task processing in terms of performance and overall processing time per configuration of the setup. The simulations provide the opportunity to identify ideal configurations for various percentages of sensitive tasks.

The code listing in Table 2 shows part of the extension of the CloudSim Plus toolkit via the `datacenterbroker` class. To distinguish between sensitive and non-senstive tasks, the `Cloudlet` class is extended to allow initialisation accordingly. To code the hybrid cloud configuration, a public data centre is created separately from a private one. Both resources are made available to the data centre broker at the start of task scheduling.

Coloured Petri Net (CPN) Implementation

The hybrid cloud is represented as a CPN model shown in Figure 5 having four integrated sections: task scheduler, log transformer, process mining monitor and business rule manager. The CPN models a process mining-influenced load-balancer for the hybrid cloud. The CPN again, allows for the modelling of both actions and states of the system [56]. The description of the parts of the CPN together with its operation are provided in more detail within Paper E. This subsection provides a brief overview of the CPN implementation of the research artefact end-product with reference to Figure 5. The operation of the load-balancer starts with the routing of tasks to either the internal datacenter

Table 2: Code Listing for the `CloudletSensitive` class which helps to instantiate and distinguish between sensitive and non-sensitive tasks [12]

CloudletSensitive Class

```
public class CloudletSensitive extends CloudletAbstract {

    private boolean sensitive;

    public CloudletSensitive(final long cloudletLength,
    final int pesNumber) {
        super(cloudletLength, pesNumber);
    }

    public CloudletSensitive(final int id,  final long cloudletLength,
    final boolean sensitive, final long pesNumber) {
        super(id, cloudletLength, pesNumber);
        this.sensitive = sensitive;
    }

    @Deprecated()
    public CloudletSensitive(
        final int id,
        final boolean sensitive,
        final long cloudletLength,
        final int pesNumber,
        final long cloudletFileSize,
        final long cloudletOutputSize,
        final UtilizationModel umCpu,
        final UtilizationModel umRam,
        final UtilizationModel umBw)
    {
        this(id,  cloudletLength, sensitive,pesNumber);
        this.setFileSize(cloudletFileSize)
            .setOutputSize(cloudletOutputSize)
            .setUtilizationModelCpu(umCpu)
            .setUtilizationModelRam(umRam)
            .setUtilizationModelBw(umBw);
    }

    @Override
    public String toString() {
        return String.format("Cloudlet %d", getId());
    }

    @Override
    public int compareTo(Cloudlet o) {
        return Double.compare(getLength(), o.getLength());
    }

    public boolean isSensitive() {
        return sensitive;
    }
```

or external public cloud depending on the business constraint encountered in the process mining monitor. Each task to be processed contains a Case id c, Activity a and Timestamp t stored in the header of the packet, utilised by the load-balancer in the routing and represented in the CPN as *(c,a,t)*. After the task is routed by the load-balancer, a log entry *(c,a,t, r)* is made with r indicating the route on which the task was sent. The log transformer formats the log data for input into the process mining monitor which in turn checks for

Table 3: Code Listing for the `DatacenterBrokerSensitive` class which helps to manage the resources and coordinate the task scheduling with respective their sensitivity [12]

DatacenterBrokerSensitive Class

```
public class DatacenterBrokerSensitive extends DatacenterBrokerAbstract {

    public DatacenterBrokerSensitive(CloudSim simulation) {
        super(simulation);
        setDatacenterSupplier(this::selectDatacenterForWaitingVms);
        setFallbackDatacenterSupplier(
        this::selectFallbackDatacenterForWaitingVms);
        setVmMapper(this::selectVmForWaitingCloudlet);
    }

    protected Datacenter selectDatacenterForWaitingVms() {
        return (getDatacenterList().isEmpty() ? Datacenter.NULL :
        getDatacenterList().get(0));
    }

    protected Datacenter selectFallbackDatacenterForWaitingVms() {
        return getDatacenterList().stream()
            .filter(dc -> !getDatacenterRequestedList().contains(dc))
            .findFirst()
            .orElse(Datacenter.NULL);
    }

    protected Vm selectVmForWaitingCloudlet(Cloudlet cloudlet) {
        if (cloudlet.isBindToVm() && getVmExecList()
        .contains(cloudlet.getVm())) {
            return cloudlet.getVm();
        }

        return getVmFromCreatedList(getNextVmIndex(
        ((CloudletSensitive)cloudlet).isSensitive()));
    }

    private int getNextVmIndex(boolean sensitivity) {
        if (getVmExecList().isEmpty()) {
            return -1;
        }
        final int vmIndex = getVmExecList().indexOf(getLastSelectedVm());

        int vmListSize = getVmExecList().size();
        if (sensitivity) {
            int i = 0;
            int count = 1;
            do {
                i = (vmIndex + count) % vmListSize;
                if (getVmExecList().get(i).getHost().getDatacenter()
                .getName().toLowerCase().contains("datacenterprivate"))
                    return i;
                count++;
            }
            while (count <= vmListSize);
        }
        return (vmIndex + 1) % vmListSize;
    }
```

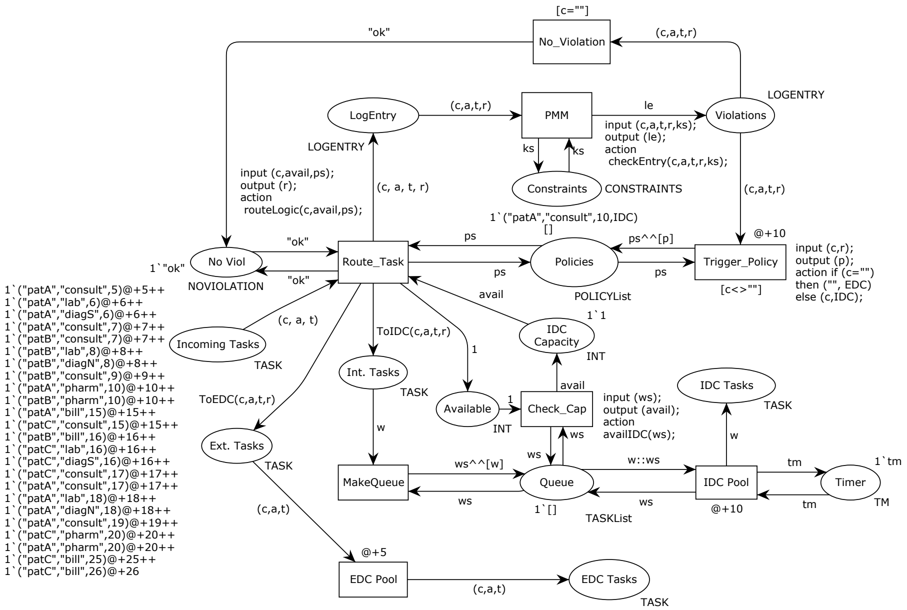


Fig. 5: CPN Model of the process mining-influenced load-balancer comprising routing, process mining and VM Pool components (culled from Paper E)

conformance of the task to the business constraint(s) specified. Violations are passed on to the business rule manager to trigger a new policy for controlling the load-balancer.

In the CPN representation, the `Incoming Tasks` place is the entry point of the system and going through the `Route_Task` transition for routing to be done. The initial marking, M_0, of the `Incoming Tasks` place is

$$M_0(Incoming\ Tasks) = [Task\ List]$$

where $Task\ List = (c, a, t) \in CASEID \times ACTIVITY \times TIME$ represents the starting tokens in the place. The tokens represent data of type `TASK` and each represented as a cartesian product of types $CASEID, ACTIVITY and TIME$. Therefore from Figure 5,

$$M_0(IncomingTasks) = 1`(``patA'', ``consult'', 5)@+5, ..., 1`(``patC'', ``bill'', 26)@+26$$

where for the first token in the list *patA* and *consult* are case id and activity respectively, occurring at timestamp 5. $M_0(Int\ Tasks) = \emptyset = M_0(Ext\ Tasks)$ is the initial marking for the respective indicated places since at the start no task routing has been executed.

The `Constraints` place has one token having data $1`(``patA'', ``consult'', 10, IDC)$ to indicate all tasks involving *patA* after timestamp *10* are to be routed to and processed in the internal data centre, *IDC*. The process mining monitor `PMM` transition determines if the log entry has violated (or not complied with) the constraint. The output token of `PMM` is a log entry with or without the case id to indicate *violation* or *no violation* respectively. `Trigger_Policy` transition consumes the token if it has a case id and outputs a *policy* token into the `Policies` place, adding to the policy list. In executing the `Route_Task` transition, the task token is sent to the `Int. Tasks` or the `Ext. Tasks` place based on the outcome of the `routeLogic` function. The availability of processing capacity of the internal datacenter is indicated by a token in the `IDC Capacity` place and is a condition for sending tokens to the `Int. Tasks` place. Tokens end up in either the `IDC Tasks` place or the `EDC Tasks` place depending on the nature of tokens in the input of `Route_Task` transition.

Table 4: Overview of how the papers relate to the methodology adopted in the study

Paper #	Methodology Stage	Outcome/Result
Paper A	Problem Definition	The need for a hybrid cloud to respond to business rules established.
Paper A & B	Requirements Specification	Hybrid cloud requirements of selected hospital described.
Paper B	Systems Analysis	Literature review on related works to establish the framework within which system will be implemented.
Paper C	System Design	Outline of how to integrate the load-balancing system with process mining in the hybrid cloud.
Paper D	System Building	Implementation of framework for integrating the process mining component into task scheduling in the hybrid cloud.
Paper E	System Building	Implementation of a model of the load-balancing and process mining components to facilitate measurement of relationships between selected metrics.

4.2 Summary of Published Articles

The published articles present the architectural overview of the hybrid cloud in Paper A; a literature review of motivations for hybrid cloud adoption in Paper B; an implementation of load-balancing using OpenStack Octavia in Paper C; an implementation of sensitive task scheduling using the CloudSim Plus tool in Paper D; and modelling of the a load-balancer for the hybrid cloud using CPN tools [57] in Paper E.

The papers together contribute to answering the research question of whether an adoption strategy of the hybrid cloud affects conformance of data processing to specified business rules. The study follows multiple methods [51, 86, 118] in attempting to answer the question, producing a series of artefacts that facilitate the validation of findings. Stages in the multiple methodology are applied both separately in each paper and also in a combined form to bring 'completeness' to the paper. Each paper also produces one artefact that acts as an input to the next stage in the methodology. Applying stages of design science methodology within the information systems research paradigm results in the sub-studies on the Problem Definition, Requirements Specification, Systems Analysis, System Design and System Building. Table 4 shows an overview of how each paper relates to the overarching methodology.

Paper A covers the investigation into the problem definition and the specification of requirements. The case of the selected hospital is employed to elicit the metrics to be measured on the performance of the proposed hybrid cloud design. Paper A delivers the description of the metrics and the initial proposed design culminating in the exploration of the definition of the research problem, requirements specification and some elements of the system design. Paper B tackles the systems analysis stage of the methodology by attempting to provide a detailed literature review on the hybrid cloud adoption strategies. The results of the analysis provides an overview of the thematic areas of business needs for which the hybrid cloud is adopted. The findings from Paper B also provide the framework within which the proposed hybrid cloud should function and answer to the thematic areas established. System Design follows next in the adopted methodology and is split into two tracks: load-balancing and task scheduling. Paper C presents the design of a process-mining controlled load-balancer using system logs obtained from the selected hospital case. The system design in Paper C covers the integration of the OpenStack Octavia [108] project. The design of the task scheduler is presented in Paper D, proposing algorithms to distinguish between sensitive and non-sensitive tasks. The design for the task scheduling system targets the CloudSim Plus [39] simulation tool for Cloud Computing. Together, Paper C and D present a system design for a hybrid cloud implementation that is data- and process-aware. The system building stage of the methodology focuses on the implementation of the working prototypes of an integration between process mining, load-balancing and the business policies or rules API engine, as outlined in Paper C and E. The system built in Paper D utilises the simulation framework that allows for inexpensive experimentation, to test out the proposed algorithms. The analysis of the system and its design characteristics provide input data for the system building stage. This system building stage is one that implements an artefact meeting the specification requirements and solving the problem defined in the methodology's first stage. The set of papers attempt to incrementally build on each other towards implementing a model that can be used to establish the relationship between the metrics and business constraints imposed on the system.

4.3 Paper A: Hybrid Cloud for Healthcare Data Sharing and Mobile Access: An Architectural Overview

This paper seeks to identify a hybrid cloud architecture that fits the adoption goals of the selected case. Among the adoption goals is a mobile-first access as a strategy in a region where mobile penetration outstrips fixed broadband Internet access. The authors point out the limitation of the typical mobile device as including limited power in terms of both energy and processing capacity, and attribute the growth in cloud computing to the increasing need to improve the user experience on the mobile device. Though many industries could benefit from the mobile-first strategy, the paper focuses on the healthcare industry because of the advantages it wields in the selected use case, situated in a region of high mobile penetration rate and low level of fixed broadband infrastructure. The authors therefore examine the practical challenges of supporting mobile-first adoption goals in a hybrid cloud and also specify business constraints pertaining to the operation of the healthcare information system. Their study seeks to measure the quality of requirements and attributes of the business process executed in the hybrid cloud in terms of availability, throughput, reliability and response time.

In related works, [9, 129] propose cloud bursting when the private cloud capacity has reached its set limits. However the data communication costs with the public cloud make the cloud bursting algorithms unsuitable for the region of the case study on account of the high costs and low reliability of the bandwidth. Apart from bursting, the cold archiving of healthcare records in the public cloud has the potential of creating unsustainable costs for a healthcare provider with relatively expensive Internet bandwidth costs. This cost can be minimised when the hybrid cloud is employed and the data is stored on the side of the infrastructure that minimises the need for bandwidth consumption. Likening the cloud to a giant computer, the cost of moving data from permanent storage to memory (RAM) for processing should not be expensive and thus a hybrid cloud is not efficient if it has data stored in a part of the cloud that causes expensive data transfer for the purpose of processing.

The paper in this wise considers the functional and technical requirements of the hybrid cloud infrastructure. The functional requirements focus on the physical architectural attributes that contribute to the realisation of the business goals. The non-functional requirements relate to the quality of the architectural attributes that are critical to optimise the usage of the hybrid cloud. For the business case selected, the paper analyses the proposed architecture based on the quality attributes of throughput, bandwidth and CPU utilisation, and response time, among others. The paper employs the associated metrics of MTTR ("Mean Time To Recovery"), MTBF ("Mean Time Between Failures") and RTO ("Recovery Time Objective") [16] to measure the extent of availability of the combined infrastructure. The article presents the case of the selected hospital putting forth the requirements of the hybrid cloud that closely supports the various operations of the organisation. The specifications target high availability at minimal costs as an important requirement in adopting the hybrid cloud for the hospital. The other specifications are: reliability, measured as MTBF/(MTBF + MTTR) $< 95\%$; response time, computed as ART = TRT + NRT, ART=Application Response Time, TRT = Transaction Response Time, NRT=Network Response Time; and throughput measured as data transmission per second in megabits.

4.4 Paper B: Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis

This paper analyses the concepts of adopting hybrid cloud and unpacks the motivations into thematic areas. The focus of the paper attempts to synthesise a collection of 34 papers that study hybrid cloud adoption and the strategies for achieving their adoption goals. The paper looks at the reasons behind the adoption with an enterprise or organisation to classify thematic areas for which hybrid cloud is employed as a strategy for solving networked computing challenges. In related works of systematic reviews, the papers identify challenges confronting hybrid cloud adoption as: network security in the public portion of the hybrid cloud; efficient management of the combination of cloud deployment types; and the complexity involved in the integration. Other related works mention the relatively high likelihood of unsecured information processing in the hybrid cloud as a major concern for adopters. The related works however stop short of presenting solutions to the cloud computing challenges employing the hybrid cloud as a strategy. The paper collates the challenges into thematic areas and describes how the adoption of the hybrid cloud helps to address associated challenges. Three thematic areas emerge from the systematic synthesis of related works bordering on performance issues, economics and security aspects. The method of inclusion of articles on hybrid cloud provided the criteria to ensure that the adoption was a result of a strategy to address computing challenges or to enhance normal enterprise systems. The criteria also ensures the hybrid cloud is employed in an experiment or study which is feasible only with the acquisition of a combined cloud infrastructure. This filters the studies that employ the combined infrastructure as a strategy to alleviate some specified challenge of adoption. The paper collates such works to establish thematic areas that are being tackled by academia and industry.

The 34 papers give three broad areas for the application of the hybrid cloud, as a strategy in the enterprise cloud computing endeavours. The thematic areas serve also as reasons for the adoption of the hybrid cloud. The largest category deals with the optimisation of the cloud resources to enhance availability and reduce latency for various kinds of workloads and improve response to various demands of users. A second largest theme dwells on the challenges of maintaining privacy in cloud adoption. Being one of the main reasons for maintaining an on-premises datacentre or a private cloud, keeping confidential information from the public cloud appears to be a major security concern for large enterprises that have many sites geographically apart. The third theme employs the hybrid cloud as a platform for minimising costs of adoption over the long term and therefore serves as the preferred model for mathematical analysis of the balance in resources between the different underlying cloud infrastructure. The results of the meta-analysis show common threads in the usage of the hybrid cloud as a cloud computing strategy

4.5 Paper C: Load balancing in hybrid clouds through process mining monitoring

Paper C proposes a task processing algorithm, regulated with process mining in a combined underlying cloud infrastructure, typically composed of a private on-premises cloud infrastructure and one or more public cloud deployments. The paper attempts to answer the question of data processing in the combined cloud infrastructure in consonance with the original adoption goals. The common adoption goals for the hybrid cloud include having maximum control of processing and data privacy whilst enjoying elastic on-demand scalability of resources. Any form of scheduling in this hybrid infrastructure should therefore comply as much as possible to the original intent of the adoption by partitioning where necessary the processes and data. The paper states that data privacy is one concern for the adoption that cannot be overlooked in scheduling tasks in the cloud. Algorithms that are able to schedule tasks towards compliance to business constraints make use of properties of the data itself. Such business constraints however are often dynamic in nature in response to customised requests by a varied set of users and processes. Programming such algorithms beforehand is therefore nearly impossible on account of the myriad pathways possible in procedurally following the logic of execution that covers the typical business model. This calls for the need to introduce an algorithm that can schedule tasks to comply dynamically to an adopter's data processing constraints. However the making and testing of such as algorithm would involve a lot of cloud resources in order to experience an appreciable level of conclusion and verification.

Paper C adopts the simulation approach to mitigate the expense of using live systems that are prorated. Simulation enables the mimicking of real-world infrastructure and scenarios. It also enables the testing and verification of scheduling algorithms that would otherwise be expensive to test because of the sheer volume of resources needed to run such algorithms. Since the algorithm is to respond to changes in the level of user requests in order to schedule the tasks in an appropriate VM, it needs to evaluate the sequence of previously occurring event data. The process of analysing the sequence of event data is delegated to a process mining tool which has the capacity to facilitate the classification of tasks as either sensitive or non-sensitive during the scheduling. Paper C employs MoBuCon [81] in this endeavour and makes use of Declare [89], a declarative visual modelling language for specifying acceptable activity sequences in a process. The visual nature of Declare facilitates user friendliness of the overall business constraint specification that can be transformed into a formal notation executable by a Prolog engine. The output of the execution represents a logical evaluation of the event data as to whether it follows the specified Declare constraint or otherwise. The dynamic execution of the MoBuCon tool culminates in a monitoring process for the simulation facilitating the determination of what constitutes sensitive data and what is considered non-sensitive. The monitoring process places the state of compliance, upon discovering of a matching sequence, into a pending state and subsequently into a satisfied or violated state depending upon its evaluation of the next sets of event data. The outcome of this monitoring process serves as an input to the scheduling algorithm and influences the allocation of tasks to VMs.

4.6 Paper D: Process Mining-constrained Scheduling in the Hybrid Cloud

This paper proposes a load-balancing approach to business rule compliance. It defines business rules as constraints that impact on the profitability or achievement of some other organisational goals. The paper acknowledges that data is increasingly being utilised in business operations and its collection, processing and storage need to be streamlined in accordance to organisational policies. The Cloud Computing paradigm, which facilitates these operations especially for multi-located offices of an organisation, introduces security concerns in the situation of public cloud adoption to augment private cloud resources. The solution of overcoming these concerns and limitation involves the combination of the differing underlying deployment models to take advantage of the benefits in each design. The resulting hybrid cloud however exacerbates the challenges of administering the cloud resources and maintaining compliance to an organisation's data processing constraints.

The paper makes use of the OpenStack Octavia load-balancer and a process mining monitoring tool to increase 'visibility' in the management of a hybrid cloud. The policy-based load balancing-as-a-service, LBaaS, built into Octavia, facilitates the specification of business constraints to control the routing of data or network packets in the hybrid cloud. Featuring prominently in the Octavia LBaaS project is the networking layer 7 (L7) policy specifications for controlling the load-balancer. These L7 policies and rules enable the load-balancer to be dynamically programmed upon the occurrence of some sequence of events. A process mining framework in turn is able to monitor event data, as they occur to determine compliance or otherwise to some specified rules.

Paper D employs an extended version of the MoBuCon (Monitoring of Business Constraints) Framework to determine compliance through the mining of the event log generated by the load-balancer. In making the event log available, Logstash facilitates the integration of the process mining tool with the load-balancer engine by transforming the output of the latter as an input for the former. Logstash also has the capability to integrate several logs where there are multiple load-balancers. The main action point in dynamically programming the load-balancer for routing at layer 7 of the networking stack, lies in the application of policies and rules at the networking layer. The Octavia L7 policy comprises of data processing principles that indicate an action to perform whenever networking packets matching rules are encountered. The policies are instantiated on trigger by the process mining monitor detecting a violation occurrence. The rules themselves are simple instructional statements that enable a logical comparison between network data packet headers and specified values to determine compliance or otherwise. If there are multiple rules in the policy, they are logically applied together and the policy action followed only if all the rules are satisfied.

In paper D, the scenario of a hospital information system is presented, where consultations are processed in a private portion of the hybrid cloud for data classified as highly sensitive. Other less sensitive data are processed without location constraints, as resources become available. The load-balancer inspects the header of the data to determine its classification and routes the data to the appropriate pool of VMs, located either in the public or private cloud. The process mining monitor not only checks for compliance in the underlying user application, but also compliance by the load-balancer itself in routing data. Where data is routed incorrectly to violate specified business constraints, the creation of a new L7 policy is triggered to realign dynamically the operations of the load-balancer and reduce the occurring violations to an acceptable threshold. From the experimental setup using synthetic data, whenever violations occurred, there was a corresponding creation of an L7 policy to effectively route the data packet to the internal data centre. The average reaction time in responding to the violation is 12.27 secs, however this is in response to the violation of one business constraint. The situation for multiple business constraints as would be experienced in the real-world is yet unexplored.

4.7 Paper E: Modelling and Simulating a Process Mining-Influenced Load-Balancer for the Hybrid Cloud

Paper E is motivated out of the need to describe more precisely, the operations of a load-balancer applying process mining and L7 policies. The choice of framework is the Coloured Petri Net (CPN) which has the features to robustly represent both mathematically and visually the flow of data in the load-balancer. Paper E extends Paper D with a modelling and simulation framework for the integration of OpenStack Octavia, Logstash and MoBuCon to produce a process mining-influenced load-balancer. The CPN model of the load-balancer facilitates the measurement of the selected metrics and the easy variation of the experimental setup in order to analyse the dynamics of the integration, especially the effect of the process mining component on the load-balancing system. The load-balancing system is highly parallelisable whilst the processing mining component is not easily parallelised and therefore has the potential to negatively impact on the processing efficiency of the integrated infrastructure. The metric measurements are founded on tasks of varying percentage sensitivity and on event logs generated by the load-balancer. The outcome of the process mining is with regards to the conformance of a sequence of system actions to specified rules, most often business related.

Per the business rule, the load-balancer is programmed to route confidential tasks to a private cloud deployment model, such as the on-premises cloud or internal data center (IDC), and thereby maintain the desired privacy for business data deemed to be sensitive. The setup of the CPN input data varies the percentage sensitivity of tasks in order to measure their effect on the metrics of processing times, queue lengths, throughput, bandwidth costs and distribution of tasks in the hybrid cloud. Paper E sets about to measure these metrics by creating a CPN model and providing the input data, business constraint(s) and the initial event log. The CPN setup features mechanisms for modelling the process mining component, the log transformation agent (Logstash), the Octavia API for L7 Policies and the load-balancer (HAProxy). Also, the CPN model sets the outcome of processing in one segment as in-coming data for the next segment in the tool-chain. The data flow among the components is governed by mathematical expressions that define the characteristic behaviour of each component, providing a robust model that precisely describes the process mining influenced load-balancer. In measuring the metrics, the mathematical expressions incorporate variables that are compared with each other to observe the trends and dynamics of the system as tasks (data), of varying percentage sensitivity, are input into the load-balancer.

The results of the simulation experiments show that the percentage of sensitive tasks has an impact on the afore-mentioned selected metrics in the hybrid cloud. Processing time generally increases with increasing percentage sensitivity of tasks. The observation of this metric trend is due to the constraint of scheduling sensitive tasks for routing only to the private portion of the hybrid cloud. Therefore increasing the workload on the same number of VMs in the private cloud capacity tends to increase the overall processing time. For the QoS requirements to be met, the number of VMs in IDC must be set to accommodate the percentage of sensitive tasks marked for the processing period. From eight VMs upwards, the simulations showed no significant change in the processing time for any given percentage of tasks.

The queue similarly adds on in length as the percentage of classified data expands. The length of the queue in the IDC pool of VMs determines whether its maximum capacity has been reached, to signal the load-balancer to route further tasks to provisioned infrastructure in the external data center (EDC) and outside the more controlled confines of the internal data center (IDC). The routing of sensitive tasks to the public cloud constitutes violations, a metric that is also considered in order to optimise the IDC VMs to accommodate the percentage of sensitive tasks within the processing period.

The bandwidth costs result from the routing of tasks to the EDC VM pool in the public cloud. The greater the percentage of tasks routed to the external data center, the higher the overall bandwidth expenditure for a given quantity of workload. The internal data center is optimised to mitigate the demand for routing tasks to the external cloud as much as possible without compromising the QoS requirements. The simulation time significantly decreases after eight VMs for 100% task sensitivity. However, beyond 32 VMs the percentage task sensitivity has insignificant impact on the processing time.

The metric of throughput measures the overall number of tasks processed per time both in the IDC and EDC VM pools. The throughput varies with changes in the number of IDC VMs: it falls as the number tasks routed to the IDC rises. The optimal range for percentage number of tasks is between 20 - 40% for any number of IDC VMs.

Discussion of Results

5 Discussion

This section attempts to synthesise the individual published articles' results. It also discusses the fulfilment of the study objectives and seeks to analyse the potential implications for hybrid cloud adoption and management.

Paper A proposed an architecture that was tested for a period of two months and the metrics of Mean Time To Failure (MTTF), Mean Time Between Failures (MTBF) and the Mean Time To Recovery (MTTR) [16] served as indicators of the availability of the hybrid cloud. The specific requirements for the selected case study are to have personally identifiable information processed in the private portion of the hybrid cloud, and have offloading capabilities for the mobile devices. The availability of the hybrid cloud was to be above the required SLA threshold of 99.5%. The results showed the nominated hybrid cloud architecture served the requirements, delivering a measured availability of 99.93%. It indicates that the hybrid cloud architecture served the business goals by offloading excess requests to the public cloud whilst guaranteeing that classified workloads are processed in the private data centre. The answer to research question RQ1 is attempted in this paper and the diagram (Figure 5 in Paper A) also shows a specific design that achieves the near-optimal solution for the selected case study. Paper A shows that, per the measured availability of the hybrid cloud, the architecture was adequate with respect to the specified business constraints.

The literature review, on motivations for hybrid cloud adoption in Paper B, resulted in three thematic areas that included privacy and security, matching the motivation for the selected organisational analysis. The outcome is safely generalisable since the requirements in the case of the selected hospital was fulfilled and the motivation for such requirements was thematic, from the hybrid cloud adoption survey of literature. The thematic areas also confirm the need to adopt the hybrid cloud deployment model as a solution to business problems and therefore the solution adopted by the selected case study falls in line with literature and general practice. For research question RQ2, Paper B introduces the areas of application data processing in the hybrid cloud and categorises them into the identified thematic areas.

The implementation of the load-balancer in Paper C resulted in the successful partitioning of data processing in the hybrid cloud and also demonstrated the successful incorporation of the process mining framework into hybrid cloud monitoring. The concept of using event logs to guide operations in the hybrid cloud enables a dynamic and robust system to be implemented to overcome the need for frequent reprogramming of partitioning algorithms. The overall aim was to enhance compliance to a set of business rules during data processing and minimise or avoid altogether violation of the business rules. The proposed framework of using process mining succeeded, to trigger API calls in the OpenStack Octavia LBaaS project and effectively influenced the load-balancer to partition data processing according to the given business rules. Paper C fulfils partially the PhD study objective of discovering organisational processes that affect the distribution of data in the hybrid cloud. This is achieved to some extent because the process mining component detects the sequence of activities that determine whether a task is sensitive or otherwise based on the business rules. Study objectives (ii) and (iii) are also partially fulfilled in Paper C where the hybrid cloud locations for storage and processing of data can be determined via process mining influencing of the load-balancer. Paper C also attempts to address research questions RQ2 and RQ3 by demonstrating batch data processing.

The results of Paper D provides further insight into the partitioning mechanism in the hybrid cloud and its impact on cloud resource provisioning. The scheduling system makes a decision based on the event logs and gives priority to accurately placing the data at the expense of speedy processing. This is to ensure the best possible compliance to the governing business rules and minimisation of the bandwidth costs. The findings in Paper D fulfil all the study objectives by strictly processing and placing data in the hybrid cloud and by showing how the varying proportion of sensitive tasks affects their processing and placement. This result is consistent with that obtained in Paper C when considering the partitioning of tasks using the process mining framework. The significance of Paper D resides in finding the appropriate balance in provisioning internal or private data centre VMs and other resources to accommodate dynamic business constraints. This is important especially for data centre planning in healthcare institutions that want to adopt cloud computing with compliance to regulatory requirements. Finally, the simulation setup in Paper D provides a broad platform for the experimentation across various domains, such as the financial services sector, making the framework generalisable for adopters of the hybrid cloud. To help address research questions RQ2 and RQ3, Paper D results show measures of various aspects of performance impact on data processing.

The results in Paper E augment that in Paper C and present a mathematical model of a load-balancer constrained by process mining. The OpenStack Octavia setup is modelled and simulated using the CPN tools and the metrics evaluated from the mathematical perspective as much as possible. The results show the mathematical relationship between the metrics on overall processing time of tasks, the queue length in the internal portion of the cloud infrastructure, the throughput of the entire system, the distribution of tasks in the cloud, the bandwidth costs and the percentage of sensitive tasks or business constraints. Determining the right balance between processing time and number of internal data centre VMs per percentage of sensitive tasks is highly desirable and the results in Paper E deliver this objective in a cost-effective solution. This fulfils the PhD study objectives with respect to determining the placement of data in the hybrid cloud to meet organisational goals and at the same time meet regulatory requirements. The CPN provides a robust mathematical foundation that enables the setup to be described precisely and implemented in other domains.

Conclusion

6 Conclusion

This section summarises the reflections on the overall PhD study, the limitations of the study and the conclusions contributed from each of the enclosed papers.

6.1 Background

The PhD study aimed to investigate the effect of a hybrid cloud adoption strategy on the conformance to business constraints imposed on data processing. The study has considered the case of a healthcare provider and employed event data culled from the information system logs. Based on the investigations culminating in the five papers enclosed, it can be concluded that the composition of the hybrid cloud architecture affects the conformance of data processing to imposed business rules, and is therefore an important factor to consider when adopting a hybrid cloud to achieve some organisational goal. The results indicate that, the greater the number of business constraints pertaining to privacy and security, the higher the capacity of the private cloud needed to ensure there is conformance during data processing.

6.2 Papers Produced and Effectiveness of Applied Methods

The study also explored the possibility of distributing hybrid cloud task processing and storage via the framework of process mining. The enclosed papers:

(i) explored a proposed hybrid cloud architecture for a selected hospital as a case study. It becomes apparent that the architectural composition of the combined cloud infrastructure has to be set out with consideration for the pertaining business constraints on data processing in order to satisfy the adoption goals.

(ii) surveyed literature for motivations in hybrid cloud adoption for fulfilling business strategies. The paper concluded that adoption strategies can be classified into three thematic areas and that the desire of the selected hospital case to adopt a hybrid cloud was well within the thematic strategies found in literature.

(iii) implemented and presented a process mining-constrained load-balancer. The paper delved into the initial design and concluded that the process mining feature was important to dynamically partition data processing in the hybrid cloud and ultimately minimise or avoid violations to the business rules.

(iv) built a task scheduling system with process mining-influenced partitioning algorithms. After successfully simulating the hybrid cloud setup, it can be concluded that the tool facilitates adequate low-cost experimentation with respect to the reigning business constraints and leads to a near-optimal and goal-attaining physical implementation. And

(v) constructed a coloured Petri net to model and simulate a load-balancer in the hybrid cloud for the purposes of mathematically evaluating the data distribution metrics. It can be concluded from this paper that there is a statistical association between the business constraint count, amount of data being processed and the capacity of the VM in the private portion of the hybrid cloud. The relationship varies with the nature of the business constraint and the CPN serves as a tool to allow such relationships to be explored cost-effectively, for decision-making in the hybrid architectural composition.

This study has modelled and simulated a process mining-influenced load-balancer by adjusting the proportions of task sensitivity on one hand and the number of VMs in the internal data centre or private cloud, on the other hand. The simulations revealed a direct relationship between the number of business constraints and the quantum of classified data; and the more elevated the proportion of information considered as sensitive, the more private cloud VMs are required to process incoming tasks in order to satisfy the business constraints and SLAs.

One avid adopter of the hybrid cloud is the hospital setting that needs to process patient bills periodically. Again, the selected hospital, in a time of epidemic, may need more data centre capacity to maintain QoS requirements. Our simulation setup, founded on the analysis of the hospital's system and event logs, helps to successfully determine the quantum of cloud infrastructure to budget, for meeting the QoS requirements and for complying with the regulatory conditions related to data privacy.

6.3 Summary of Contributions and Future Work

The augmentation to the body of scholarship by this PhD study can be outlined as:

(i) Introduction of the specific tool-chain concept consisting of the load-balancer, log transform agent, process mining agent and business policy API for partitioning the processing of tasks in the hybrid cloud.

(ii) Implementation of the tool-chain framework using the open source OpenStack cloud system to offer real-world integration.

(iii) Implementation of the tool-chain framework using open source CloudSim Plus tool to offer avenues for cost-effective simulation of task scheduling in a hybrid cloud.

(iv) Implementation of the tool-chain framework using CPN tools to facilitate a concise, mathematical perspective of metric measurements.

(v) Establishment of three thematic areas in a literature survey to broaden knowledge in hybrid cloud research and also help shape business decisions during adoption.

(vi) Refining the network requirements for healthcare providers with respect to hybrid cloud adoption especially in regions of under-developed Internet and Cloud infrastructure.

Future work to be considered centre around applying the framework generally in edge-computing where off-loading of process- and data-intensive tasks can be dynamically influenced by event data rather than being pre-programmed.

References

[1] W. Aalst, “Process Discovery: An Introduction,” in *Process Mining.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 163–194. [Online]. Available: http://link.springer.com/10.1007/978-3-662-49851-4{_}6

[2] H. Abrishami, A. Rezaeian, and M. Naghibzadeh, “A novel deadline-constrained scheduling to preserve data privacy in hybrid Cloud,” in *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2015, pp. 234–239.

[3] R. Accorsi, M. Ullrich, and W. M. P. Van Der Aalst, “Process mining,” *Informatik-Spektrum*, vol. 35, no. 5, pp. 354–359, oct 2012. [Online]. Available: http://link.springer.com/10.1007/s00287-012-0641-4

[4] M. S. Aktas, “Hybrid cloud computing monitoring software architecture,” in *Concurrency Computation*, vol. 30, no. 21, nov 2018, p. e4694. [Online]. Available: http://doi.wiley.com/10.1002/cpe.4694

[5] M. Alcaraz-Mejia, R. Campos-Rodriguez, and M. Caballero-Gutierrez, “Modeling and Simulation of Task Allocation with Colored Petri Nets,” in *Computer Simulation*, D. Cvetkovic, Ed. InTech, jun 2017, ch. 10, pp. 213 – 253. [Online]. Available: http://dx.doi.org/10.5772/67950

[6] M. Alrokayan, A. V. Dastjerdi, and R. Buyya, "SLA-Aware Provisioning and Scheduling of Cloud Resources for Big Data Analytics," in *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2014, pp. 1–8.

[7] J. Altmann and M. M. Kashef, "Cost model based service placement in federated hybrid clouds," *Future Generation Computer Systems*, vol. 41, pp. 79–90, dec 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X14001629

[8] K. Annapureddy, "Security Challenges in Hybrid Cloud Infrastructures," *Aalto University, Seminar on Network Security*, pp. T–110.5290, 2010. [Online]. Available: http://www.cse.hut.fi/en/publications/B/11/papers/annapureddy.pdf

[9] D. R. Avresky, P. Di Sanzo, A. Pellegrini, B. Ciciani, and L. Forte, "Proactive scalability and management of resources in hybrid clouds via machine learning," in *Proceedings - 2015 IEEE 14th International Symposium on Network Computing and Applications, NCA 2015*. IEEE, sep 2016, pp. 114–119. [Online]. Available: http://ieeexplore.ieee.org/document/7371712/

[10] M. M. Ayash, "Research Methodologies in Computer Science and Information Systems," 2013. [Online]. Available: https://pdfs.semanticscholar.org/3d85/8e3df0997f3fdaacab386c4eae9117e90ccb.pdf?{_}ga=2.45336871.246332989.1592765837-724681668.1591217478

[11] K. K. Azumah, L. T. Sørensen, and R. Tadayoni, "Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis," in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*. IEEE, aug 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8506887/

[12] K. K. Azumah, S. Kosta, and L. T. Sørensen, "Scheduling in the hybrid cloud constrained by process mining," in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, vol. 2018-Decem. IEEE, dec 2018, pp. 308–313. [Online]. Available: https://ieeexplore.ieee.org/document/8591035/

[13] K. K. Azumah, R. Tadayoni, and L. T. Sørensen, "Hybrid Cloud for Healthcare Data Sharing and Mobile Access: An Architectural Overview," *Nordic and Baltic Journal of Information and Communications Technologies*, vol. 2018, no. 1, pp. 153,176, 2018.

[14] K. Bakshi, "Secure hybrid cloud computing: Approaches and use cases," in *2014 IEEE Aerospace Conference*, 2014, pp. 1–8.

[15] Y. Balagoni and R. R. Rao, "A cost-effective SLA-aware scheduling for hybrid cloud environment," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2016, pp. 1–7.

[16] A. K. Bardsiri and S. M. Hashemi, "QoS Metrics for Cloud Computing Services Evaluation," *International Journal of Intelligent Systems and Applications*, vol. 6, no. 12, pp. 27–33, nov 2014. [Online]. Available: http://www.mecs-press.org/ijisa/ijisa-v6-n12/v6n12-4.html

[17] R. Baskerville, J. Pries-Heje, and J. Venable, “Soft design science methodology,” in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST ’09*, ser. the 4th International Conference. New York, New York, USA: ACM Press, may 2009, p. 1. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1555619.1555631

[18] L. F. Bittencourt, E. R. Madeira, and N. L. Da Fonseca, “Scheduling in hybrid clouds,” *IEEE Communications Magazine*, vol. 50, no. 9, pp. 42–47, 2012.

[19] J. M. Bohli, N. Gruschka, M. Jensen, L. Lo Iacono, and N. Marnau, “Security and privacy-enhancing multicloud architectures,” *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 212–224, 2013.

[20] L. M. Carril, Z. Martín-Rodríguez, C. Mouriño, A. Gómez, R. Díaz, and C. Fernández, “Advanced Computing Services for Radiotherapy Treatment Planning on Cloud,” *Cloud Computing*, pp. 529–551, oct 2011.

[21] M. Carroll, A. van der Merwe, and P. Kotze, “Secure cloud computing: Benefits, risks and controls,” in *2011 Information Security for South Africa*, vol. 1-9. IEEE, aug 2011, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/6027519/

[22] A. Carugati, “Information system development activities and inquiring systems: an integrating framework,” *https://doi.org/10.1057/ejis.2008.5*, vol. 17, no. 2, pp. 143–155, 2017. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1057/ejis.2008.5

[23] J. Cerviño, F. Escribano, P. Rodríguez, I. Trajkovska, and J. Salvachúa, “Leasing Videoconference Resources on Hybrid Clouds,” *Cloud Computing*, pp. 511–527, oct 2011.

[24] P. S. Chen, C. J. Yu, and G. Y. H. Chen, “Applying Task-Technology Fit Model to the Healthcare Sector: a Case Study of Hospitals’ Computed Tomography Patient-Referral Mechanism,” *Journal of Medical Systems*, vol. 39, no. 8, aug 2015.

[25] Z. N. Chen, K. Chen, J. L. Jiang, L. F. Zhang, S. Wu, Z. W. Qi, C. M. Hu, Y. W. Wu, Y. Z. Sun, H. Tang, A. B. Sun, and Z. L. Kang, “Evolution of Cloud Operating System: From Technology to Ecosystem,” *Journal of Computer Science and Technology*, vol. 32, no. 2, pp. 224–241, mar 2017.

[26] F. Chesani, A. Ciampolini, D. Loreti, and P. Mello, “Map reduce autoscaling over the cloud with process mining monitoring,” in *Communications in Computer and Information Science*, vol. 740. Springer, Cham, apr 2017, pp. 109–130. [Online]. Available: http://link.springer.com/10.1007/978-3-319-62594-2_6

[27] N. Chopra and S. Singh, “Survey on scheduling in hybrid clouds,” in *5th International Conference on Computing Communication and Networking Technologies, ICCCNT 2014*. IEEE, jul 2014, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6963050/

[28] F. J. Clemente-Castelló, B. Nicolae, R. Mayo, J. C. Fernández, and M. M. Rafique, “On Exploiting Data Locality for Iterative Mapreduce Applications in Hybrid Clouds,” in *Proceedings of the 3rd IEEE/ACM International*

Conference on Big Data Computing, Applications and Technologies, ser. BDCAT '16. New York, NY, USA: ACM, dec 2016, pp. 118–122. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3006299.3006329

[29] C. Cloud Standards Customer Council, "Practical Guide to Hybrid Cloud Computing," 2016.

[30] J. F. Cox and J. G. Schleier, *Theory of constraints handbook.* McGraw-Hill, 2010.

[31] E. Crookshanks, "Development Methodologies and SDLC," in *Practical Enterprise Software Development Techniques.* Apress, Berkeley, CA, 2015, pp. 37–59.

[32] I. Cushman, L. Chen, D. B. Rawat, and N.-A. Le-Khac, "Designing Hybrid Cloud Computing Framework Using OpenStack for Supporting Multimedia with Security and Privacy," in *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*, ser. MobiMedia '16. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 184–189. [Online]. Available: http://dl.acm.org/citation.cfm?id=3021385.3021418

[33] D. N. Dagbanja, "The Right to Privacy and Data Protection in Ghana," in *African Data Privacy Laws.* Springer, Cham, 2016, pp. 229–248. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-47317-8_10

[34] J. T. Davies, *The Scientific Approach*, 2nd ed. New York: Academic Press, 1965. [Online]. Available: https://philpapers.org/rec/DAVTSA-3

[35] S. De Cnudde, J. Claes, and G. Poels, "Improving the quality of the Heuristics Miner in ProM 6.2," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7678–7690, dec 2014. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0957417414003509

[36] F. de Vaulx, E. Simmon, and R. Bohn, "Cloud computing service metrics description," apr 2017. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-307.pdf

[37] P. Donadio, G. B. Fioccola, R. Canonico, and G. Ventre, "Network security for Hybrid Cloud," in *2014 Euro Med Telco Conference (EMTC)*, 2014, pp. 1–6.

[38] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications.* Springer Vienna, 2014.

[39] M. C. Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio, and M. M. Freire, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," in *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, 2017, pp. 400–406.

[40] R. Gandhi, Y. C. Hu, and M. Zhang, "Yoda: A highly available layer-7 load balancer," in *Proceedings of the 11th European Conference on Computer Systems, EuroSys 2016.* ACM Press, 2016, pp. 1–16. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2901318.2901352

[41] D. Georgiou and C. Lambrinoudakis, "GDPR Compliance: Proposed Guidelines for Cloud-Based Health Organizations," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12501 LNCS, pp. 156–169, sep 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-64330-0_10

[42] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, "An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach," *Future Generation Computer Systems*, vol. 78, pp. 191–210, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.02.022

[43] A. Q. Gill and S. Hevary, "Cloud monitoring data challenges: A systematic review," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9947 LNCS, pp. 72–79, 2016.

[44] S. Goyal, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review," *International Journal of Computer Network and Information Security*, vol. 6, no. 3, pp. 20–29, 2014. [Online]. Available: http://www.mecs-press.org/ijcnis/ijcnis-v6-n3/v6n3-3.html

[45] S. Gregor, "Design Theory in Information Systems," *Australasian Journal of Information Systems*, vol. 10, no. 1, 2002. [Online]. Available: http://journal.acs.org.au/index.php/ajis/article/view/439

[46] L. Griebel, H.-U. Prokosch, F. Köpcke, D. Toddenroth, J. Christoph, I. Leb, I. Engel, and M. Sedlmayr, "A scoping review of cloud computing in healthcare," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, p. 17, dec 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25888747http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4372226http://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-015-0145-7

[47] B. R. Grunstra, R. L. Ackoff, S. K. Gupta, and J. S. Minas, "Scientific Method: Optimizing Applied Research Decisions." *Philosophy and Phenomenological Research*, vol. 25, no. 4, p. 594, jun 1965. [Online]. Available: https://philpapers.org/rec/ACKSMO-2

[48] B. V. Halle and L. Goldberg, *Business Rule Revolution (ebook): Running Business the Right Way*. Happy About, 2006. [Online]. Available: http://books.google.es/books?id=s7AMgjYSkJIC

[49] K. Z. Haung, Y. S. Chen, and Y. K. Chung, "Perspectives on process mining within cloud computing," in *2011 3rd International Conference on Advanced Computer Control, ICACC 2011*, 2011, pp. 656–660.

[50] J. Heikka, R. Baskerville, and M. Siponen, "A Design Theory for Secure Information Systems Design Methods," *Journal of the Association for Information Systems*, vol. 7, no. 11, pp. 725–770, nov 2006. [Online]. Available: https://jyx.jyu.fi/handle/123456789/68257

[51] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly: Management Information Systems*, vol. 28, no. 1, pp. 75–105, 2004.

[52] R. Hill, L. Hirsch, P. Lake, and S. Moshiri, *Guide to Cloud Computing*, ser. Computer Communications and Networks. London: Springer London, 2013. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-4603-2http://www.springerlink.com/index/10.1007/978-1-4471-4603-2

[53] W. v. d. A. H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, "ProM 6: The Process Mining Toolkit," in *CEUR Workshop Proceedings*, vol. 615, 2010, pp. 34–39. [Online]. Available: http://www.processmining.org/{_}media/publications/verbeek2010c.pdf

[54] X. Huang and X. Du, "Achieving big data privacy via hybrid cloud," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2014, pp. 512–517.

[55] T. Jain and J. Hazra, "Hybrid Cloud Computing Investment Strategies," *Production and Operations Management*, vol. 28, no. 5, pp. 1272–1284, may 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/poms.12991https://onlinelibrary.wiley.com/doi/abs/10.1111/poms.12991https://onlinelibrary.wiley.com/doi/10.1111/poms.12991

[56] K. Jensen, "An introduction to the theoretical aspects of coloured petri nets," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 803 LNCS, pp. 230–272, 1994.

[57] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, jun 2007. [Online]. Available: https://link.springer.com/article/10.1007/s10009-007-0038-x

[58] W. Z. Jiang and Z. Q. Sheng, "A New Task Scheduling Algorithm in Hybrid Cloud Environment," in *2012 International Conference on Cloud and Service Computing*, 2012, pp. 45–49.

[59] D. Jones, S. Gregor, and T. Lynch, "An information systems design theory for web-based education," in *Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education*, 2003, pp. 603–608. [Online]. Available: https://djon.es/blog/wp-content/uploads/2008/12/isdt.pdf

[60] M. Joshi, N. Tewari, and S. K. Budhani, "Security challenges in implementing a secured hybrid cloud model for e-health services," in *Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020*. Institute of Electrical and Electronics Engineers Inc., dec 2020, pp. 3–7.

[61] J. Jung and H. Kim, "MR-CloudSim: Designing and implementing MapReduce computing model on CloudSim," in *2012 International Conference on ICT Convergence (ICTC)*, ser. 2012 International Conference on ICT Convergence (ICTC). IEEE, oct 2012, pp. 504–509. [Online]. Available: http://ieeexplore.ieee.org/document/6387186/

[62] K. Kanagaraj and S. Swamynathan, "Structure aware resource estimation for effective scheduling and execution of data intensive workflows in cloud," *Future*

Generation Computer Systems, vol. 79, pp. 878–891, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.09.001

[63] T. Karthikeyan and T. Nandhini, "Dependent component cost model of legacy application for hybrid cloud," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, 2016, pp. 1–5.

[64] M. J. Kavis, *Architecting The Cloud*, M. Kavis, Ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., jan 2014. [Online]. Available: http://doi.wiley.com/10.1002/9781118691779

[65] S. U. Khan and N. Ullah, "Challenges in the adoption of hybrid cloud: an exploratory study using systematic literature review," *The Journal of Engineering*, vol. 2016, no. 5, pp. 107–118, may 2016.

[66] M. U. Kolhatkar and V. A. Gulhane, "A review on hybrid cloud approach for sharing health information and management," in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017, pp. 1–4.

[67] Y.-H. Kuo, Y.-L. Jeng, and J.-N. Chen, "A Hybrid Cloud Storage Architecture for Service Operational High Availability," in *2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*. IEEE, jul 2013, pp. 487–492. [Online]. Available: http://ieeexplore.ieee.org/document/6605838/

[68] G. Laatikainen, O. Mazhelis, and P. Tyrvainen, "Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle," *Journal of Systems and Software*, vol. 122, pp. 180–201, dec 2016. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0164121216301613

[69] J. Lawler, A. Joseph, and H. H.-B. Howell-Barber, "A Case Study Of Determinants Of An Effective Cloud Computing Strategy," *Review of Business Information Systems (RBIS)*, vol. 16, no. 3, pp. 145–156, jul 2012.

[70] R. Lenz, R. Blaser, M. Beyer, O. Heger, C. Biber, M. Bäumlein, and M. Schnabel, "IT support for clinical pathways—Lessons learned," *International Journal of Medical Informatics*, vol. 76, no. SUPPL. 3, pp. S397–S402, dec 2007.

[71] Y. C. Liu and C. L. Li, "A Stratified Monitoring Model for Hybrid Cloud," *Applied Mechanics and Materials*, vol. 719-720, no. Materials and Engineering Technology, pp. 900–906, jan 2015. [Online]. Available: https://www.scientific.net/AMM.719-720.900

[72] C. Low, Y. Chen, and M. Wu, "Understanding the determinants of cloud computing adoption," *Industrial Management and Data Systems*, vol. 111, no. 7, pp. 1006–1023, 2011.

[73] M. S. Maaref, "Cloud computing in Africa - Situation and perspectives," *Telecommunication Development Sector - ITU*, p. 70, 2012. [Online]. Available: http://www.itu.int/ITU-D/treg/publications/Cloud_Computing_Afrique-e.pdf

[74] F. M. Maggi, C. Di Francescomarino, M. Dumas, C. Ghidini, and C. Ghidini, "Predictive Monitoring of Business Processes," in *Lecture Notes in Computer Science*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 457–472. [Online]. Available: http://link.springer.com/10.1007/978-3-319-07881-6_31

[75] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of Cloud Computing Services: A Total Cost of Ownership Approach," in *45th Hawaii International Conference on System Sciences*, 2012, pp. 1563–1572. [Online]. Available: https://www.computer.org/csdl/proceedings/hicss/2012/4525/00/4525b563.pdf

[76] E. Mashandudze and B. Dwolatzky, "Major Challenges Impeding the Fast Adoption of Cloud Computing : A Case Study of South African Organisations and Emerging Economics," *Open Journal of mobile computing and cloud computing*, vol. 2, no. 1, pp. 1–20, 2015.

[77] M. Mattess, R. N. Calheiros, and R. Buyya, "Scaling MapReduce Applications Across Hybrid Clouds to Meet Soft Deadlines," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, ser. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE, mar 2013, pp. 629–636. [Online]. Available: http://ieeexplore.ieee.org/document/6531813/

[78] C. R. May, M. Johnson, and T. Finch, "Implementation, context and complexity," *Implementation Science*, vol. 11, no. 1, p. 141, dec 2016. [Online]. Available: http://implementationscience.biomedcentral.com/articles/10.1186/s13012-016-0506-3

[79] O. Mazhelis and P. Tyrväinen, "Economic aspects of hybrid cloud infrastructure: User organization perspective," *Information Systems Frontiers*, vol. 14, no. 4, pp. 845–869, sep 2012. [Online]. Available: http://link.springer.com/article/10.1007/s10796-011-9326-9http://link.springer.com/10.1007/s10796-011-9326-9

[80] P. M. Mell and T. Grance, "The NIST definition of cloud computing," *NIST Special Publication*, vol. 145, p. 7, 2011. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[81] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. van der Aalst, "Monitoring business constraints with the event calculus," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 1, pp. 1–30, dec 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2542182.2542199

[82] A. J. A. J. Morgan, *Business rules and information systems : aligning IT with business goals*. Addison-Wesley, 2002.

[83] F. Moscato, R. Aversa, B. Di Martino, T.-F. Fortiş, and V. Munteanu, "An analysis of mOSAIC ontology for Cloud resources annotation," in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2011, pp. 973–980.

[84] M. Mujinga and B. Chipangura, "Cloud computing concerns in developing economies," in *Proceedings of the 9th Australian Information Security Management Conference*, 2011, pp. 196–203. [Online]. Available: http://ro.ecu.edu.au/ism/127

[85] E. Nagel, "The Structure of Science: Problems in the Logic of Scientific Explanation," *Philosophy*, vol. 37, no. 142, 1962. [Online]. Available: https://philpapers.org/rec/NAGTSO-13

[86] J. F. Nunamaker Jr., M. Chen, and T. D. Purdin, "Systems development in information systems research." *Journal of Management Information Systems*, vol. 7, no. 3, pp. 89 – 106, 1990.

[87] O. J. Okesola, A. A. Adebiyi, A. A. Owoade, O. Adeaga, O. Adeyemi, and I. Odun-Ayo, “Software Requirement in Iterative SDLC Model,” in *Advances in Intelligent Systems and Computing*, vol. 1224 AISC. Springer, Cham, jul 2020, pp. 26–34.

[88] K. Y. Oktay, S. Mehrotra, V. Khadilkar, and M. Kantarcioglu, “SEMROD: Secure and Efficient MapReduce Over HybriD Clouds,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’15. New York, NY, USA: ACM, 2015, pp. 153–166. [Online]. Available: http://doi.acm.org/10.1145/2723372.2723741

[89] M. Pešić and W. M. van der Aalst, “A Declarative Approach for Flexible Business,” in *Business Process Management*. Springer-Verlag Berlin, 2006, pp. 169–180. [Online]. Available: http://link.springer.com/10.1007/11837862_18

[90] F. Pianese, P. Bosch, A. Duminuco, N. Janssens, T. Stathopoulos, and M. Steiner, “Toward a cloud operating system,” *2010 IEEE/IFIP Network Operations and Management Symposium Workshops, NOMS 2010*, pp. 335–342, 2010.

[91] R. S. Pressman, *Software Engineering A Practitioner’s Approach - 6th Ed.* McGraw-Hill, New York, NY, 2005.

[92] H. U. Prokosch and J. Dudeck, *Hospital Information Systems: Design and Development Characteristics; Impact and Future Architecture, 1e (International Congress Series)*, J. D. H.U. Prokosch, Ed. Amsterdam: Elsevier, 1995.

[93] L. Qian, Z. Luo, Y. Du, and L. Guo, “Cloud computing: An overview,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5931 LNCS, 2009, pp. 626–631.

[94] S. Radack, “The system development life cycle (sdlc),” 2009-04-29 2009. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=902622

[95] H. Rahhali and Rahhali, “A new conception of load balancing in cloud computing using Hybrid heuristic algorithm,” *International Journal of Computer Science Issues*, vol. 15, no. 6, pp. 1–8, nov 2018.

[96] V. Rao, K. Naveena, R. David, and M. S. Narayana, “A new computing environment using hybrid cloud,” *Journal of Information Sciences and Computing Technologies*, vol. 3, no. 1, pp. 180–185, 2015. [Online]. Available: www.scitecresearch.com/journals/index.php/jisct/index

[97] N. Rathore, “Performance of Hybrid Load Balancing Algorithm in Distributed Web Server System,” *Wireless Personal Communications*, vol. 101, no. 3, pp. 1233–1246, aug 2018. [Online]. Available: http://link.springer.com/10.1007/s11277-018-5758-6

[98] N. Regola and N. V. Chawla, “Storing and Using Health Data in a Virtual Private Cloud,” *J Med Internet Res*, vol. 15, no. 3, 2013.

[99] P. Ross and M. Blumenstein, “Cloud computing: the nexus of strategy and technology,” *Journal of Business Strategy*, vol. 34, no. 4, pp. 39–47, jul 2013. [Online]. Available: http://www.emeraldinsight.com/doi/10.1108/JBS-10-2012-0061

[100] S. Schulte, C. Janiesch, S. Venugopal, I. Weber, and P. Hoenisch, “Elastic Business Process Management: State of the art and open challenges for BPM in the cloud,” *Future Generation Computer Systems*, vol. 46, pp. 36–50, may 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167739X1400168X

[101] S. Sharma, S. Verma, K. Jyoti, and Kavita, “Hybrid bat algorithm for balancing load in cloud computing,” *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4.12 Special Issue 12, pp. 26–29, oct 2018. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/20986

[102] S. Snedaker and C. Rima, “Business Continuity and Disaster Recovery Planning for IT Professionals: Second Edition,” *Business Continuity and Disaster Recovery Planning for IT Professionals: Second Edition*, pp. 1–577, sep 2013.

[103] B. Sosinsky, *Cloud Computing Bible.* Wiley, 2010.

[104] S. Souravlas, S. Katsavounis, and S. Anastasiadou, “On Modeling and Simulation of Resource Allocation Policies in Cloud Computing Using Colored Petri Nets,” *Applied Sciences*, vol. 10, no. 16, p. 5644, aug 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/16/5644

[105] R. Spies, S. Grobbelaar, and A. Botha, “A Scoping Review of the Application of the Task-Technology Fit Theory,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12066 LNCS. Springer, Cham, apr 2020, pp. 397–408. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-44999-5_33

[106] A. Srinivasan, Q. M. Abdul, and V. Vijayakumar, “Era of Cloud Computing: A New Insight to Hybrid Cloud,” *Procedia Computer Science*, vol. 50, pp. 42–51, jan 2015.

[107] E. Sturrus and O. Kulikova, “Orchestrating Hybrid Cloud Deployment: An Overview,” *Computer*, vol. 47, no. 6, pp. 85–87, jun 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6838945/

[108] The OpenStack Octavia Project , “Introducing Octavia.” [Online]. Available: https://docs.openstack.org/octavia/latest/reference/introduction.html

[109] ——, “Octavia API — octavia documentation.” [Online]. Available: https://docs.openstack.org/api-ref/load-balancer/index.html

[110] W.-T. Tsai, X. Sun, and J. Balasooriya, “Service-Oriented Cloud Computing Architecture,” *2010 Seventh International Conference on Information Technology: New Generations*, pp. 684–689, 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5501650

[111] N. Ullah and S. Khan, “Challenges in the adoption of hybrid cloud: an exploratory study using systematic literature review,” *The Journal of Engineering*, 2016. [Online]. Available: http://digital-library.theiet.org/docserver/fulltext/joe/2016/5/JOE.2016.0089.pdf?expires=1517258103&id=id&accname=guest&checksum=C3FBB877712DC268F3D6CC394B63045Chttp://digital-library.theiet.org/content/journals/10.1049/joe.2016.0089

[112] I. T. Union, “Study on international Internet connectivity in sub-Saharan Africa,” 2013. [Online]. Available: https://www.itu.int/en/ITU-D/Regulatory-Market/Documents/IIC_Africa_Final-en.pdf

[113] M. Unuvar, M. Steinder, and A. N. Tantawi, “Hybrid Cloud Placement Algorithm,” in *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, 2014, pp. 197–206.

[114] S. ur Rahman, “Theory of constraints: A review of the philosophy and its applications,” *International Journal of Operations & Production Management*, vol. 18, no. 4, pp. 336–355, 1998.

[115] W. van der Aalst, “Process Mining,” *ACM Transactions on Management Information Systems*, vol. 3, no. 2, pp. 1–17, jul 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2229156.2229157

[116] W. Van der Aalst, J. Desel, and A. Oberweis, *Business Process Management: Models, Techniques, and Empirical Studies*. Springer, 2000, vol. 1806.

[117] M. L. van Eck, X. Lu, S. J. J. Leemans, and W. M. P. van der Aalst, “PM

2

: A Process Mining Project Methodology,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9097, pp. 297–313, 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-19069-3_19

[118] J. Venable, J. Pries-Heje, and R. Baskerville, “A Comprehensive Framework for Evaluation in Design Science Research,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7286 LNCS, pp. 423–438, 2012. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-29863-9_31

[119] J. Walls and G. Widmeyer, “Assessing Information System Design Theory in Perspective: How Useful Was Our 1992 Initial Rendition?” *Journal of Information Technology Theory and Application JITTA*, vol. 6, no. 2, p. 6, 2004. [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1126&context=jitta

[120] S. F. Wamba and K. Michael, “An information systems design theory for an RFID university-based laboratory,” in *Information Systems Foundations: The Role of Design Science*, 2010, pp. 233–252. [Online]. Available: https://library.oapen.org/bitstream/handle/20.500.12657/33717/459290.pdf?seq#page=247

[121] B. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, “Improving task scheduling with parallelism awareness in heterogeneous computational environments,” *Future Generation Computer Systems*, vol. 94, pp. 419–429, 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.11.012

[122] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, “A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds,” *Cluster Computing*, vol. 23, no. 4, pp. 2809–2834, dec 2020.

[123] P. Weber, B. Bordbar, and P. Tino, “A principled approach to mining from noisy logs using Heuristics Miner,” in *Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, ser. 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, apr 2013, pp. 119–126. [Online]. Available: http://ieeexplore.ieee.org/document/6597226/

[124] H. Wei and F. Meng, "A Novel Scheduling Mechanism For Hybrid Cloud Systems," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, 2016, pp. 734–741.

[125] K. E. Weick, "What Theory is Not, Theorizing Is," *Administrative Science Quarterly*, vol. 40, no. 3, p. 385, sep 1995.

[126] J. Weinman, "Hybrid Cloud Economics," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 18–22, jan 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7420473/

[127] E. O. Yeboah-boateng and K. A. Essandoh, "Factors Influencing the Adoption of Cloud Computing by Small and Medium Enterprises in Developing Economies," *International Journal of Emerging Science and Engineering*, no. 4, pp. 13–20, 2014.

[128] W. P. Yuen and K. B. Chuah, "Data visibility and trust enhancement of enterprise customers in cloud computing services," *2017 12th International Conference for Internet Technology and Secured Transactions, ICITST 2017*, pp. 417–422, may 2018.

[129] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen, "Proactive workload management in hybrid cloud computing," *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 90–100, mar 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6701292/

[130] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: Privacy-aware Data Intensive Computing on Hybrid Clouds," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 515–526. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046767

[131] L. Zuo, L. Shu, S. Dong, Y. Chen, and L. Yan, "A Multi-Objective Hybrid Cloud Resource Scheduling Method Based on Deadline and Cost Constraints," *IEEE Access*, vol. 5, pp. 22 067–22 080, 2017.

Part II

Papers

Paper A

Hybrid Cloud for Healthcare Data Sharing and Mobile Access: An Architectural Overview

Azumah, Kenneth Kwame, Tadayoni, Reza and Sørensen, Lene Tolstrup

The paper has been published in the
Nordic and Baltic Journal of Information and Communications Technologies
Vol. 2018(1), pp. 153–176, 2018.

Abstract

A hybrid cloud computing architecture that places the mobile device or thin clients first is the logical choice especially in regions with low fixed broadband but high mobile penetration rates. The hybrid cloud model combines the benefits of computing resource elasticity in the public cloud whilst maintaining control of sensitive data and mission-critical applications mainly in a private cloud infrastructure. One industry that stands to benefit from extending mobile computing with hybrid cloud infrastructure is the healthcare industry where clinicians need the ability to access healthcare data from different locations and across multiple devices. This paper identifies a hybrid cloud architecture to support mobile device access and satisfy specific business requirements using the case of a selected hospital in Ghana. In the design of the hybrid cloud architecture the functional and non-functional viewpoints are considered using a case study where the selected hospital is used to conceptually define the requirements and set-up of a hybrid cloud architecture. The case study approach is used to illustrate the practical challenges and limitations for a hybrid cloud architecture in a developing country. The suitability of the design was validated using the metrics of availability, reliability, response time and throughput.

keywords

Hybrid cloud, cloud architecture, mobile cloud, cloud computing in healthcare

1 Introduction

A hybrid cloud is a combination of two or more distinct cloud deployment models where the models are either public, private or community clouds [1]. In a state-of-the-cloud survey done by RightScale[1] in January 2016, 95% of worldwide respondents - companies with over 1000 employees - were using cloud and 71% were hybrid cloud adopters [2]. According to RightScale the significant rise in hybrid cloud adoption was mainly due to public cloud users adding private cloud resource pools to their infrastructure. Gartner[2] projected the public cloud services to grow to $208.6 billion in 2016 an increase of 17.2% on 2015 with a domination of hybrid cloud computing scenarios as many traditional IT organisations continued incorporating their existing datacentres into their overall cloud adoption [3]. Cloud implementations in the developing world is also expected to grow mainly driven by the growing mobile phone penetration rates especially in sub-Saharan Africa is projected grow from 43% in 2015 to 51% of total population by 2020 [4–6] and this is signalling the need to make provision for mobile cloud during architectural planning and implementation.

[1] RightScale (http://www.rightscale.com)

[2] Gartner Inc. (http://www.gartner.com) is an IT related research and advisory firm founded in 1979.

Cloud computing wields great potential to facilitate the running of resource-intensive applications on mobile devices. Smart phones and tablets typically have limitations in memory, storage, computational power and energy capacity which negatively affect users experience when running resource-intensive tasks, specifically those that are computationally, communication or data intensive [7]. Cloud computing offers the possibility of offloading the resource intensive tasks in mobile applications for processing in the resource rich cloud and thereby improve user experience closer to that seen in the desktop computer [8, 9]. Mobile devices continue to grow more powerful in terms of processing power and memory capacity however the battery as a power source has lagged behind in development affecting the overall capacity of the mobile devices to meet user expectations in resource-intensive applications [10].

One industry that stands to benefit from extending mobile computing with cloud infrastructure is the healthcare industry [11] where clinicians need the ability to access healthcare data from different locations and across multiple devices [12]. Mobile cloud is one way by which mobile devices could be used to access and process electronic health records of patients, to view diagnostic images in various resolutions or collect health data from patients [13]. In Ghana, as in other developing countries, cloud computing and mobile applications and policies are being utilised in the health care sector to mitigate the pressure on clinical facilities and personnel [14, 15]. With the promising potential of cloud computing in developing economies, studies [16–19] have been done to help shape cloud adoption but the appropriateness of a hybrid cloud architecture to meet specific business requirements have received little attention. In the healthcare industry, specific network requirements of availability [12, 20, 21], economic archive storage capacity [20, 22], mobile device accessibility [12, 13], regulatory compliance [20, 23] and security [24] demand particular characteristics [25] in the hybrid cloud. From the afore-mentioned trends, a hybrid cloud computing architecture that places the mobile device or thin clients first is the logical choice especially in regions with low fixed broadband but high mobile penetration rates. Further, Griebel et. al [26] categorised articles written on MEDLINE into six topics areas that include availability, storage, mobile access, compliance and security to be high on the list of networking requirements when considering the adoption of cloud computing.

This paper examines the key components of hybrid cloud architecture, specifically a public cloud and on-premise private cloud combination, for supporting mobile device access using the case of a selected hospital in Ghana. The cloud combination affords a level of flexibility which cannot be found in either the public cloud or on-premise private cloud alone: the advantage of elastic scaling out within the public cloud with maximum control of sensitive data in the private on-premise datacentre. In the design of the hybrid cloud architecture the functional and non-functional viewpoints are considered using a case study where the selected hospital is used to conceptually define the requirements and set-up of a hybrid cloud architecture. The case study approach is used to illustrate the practical challenges and limitations for a hybrid cloud architecture in a developing country. The business case for the hybrid cloud adoption is examined and tested against the selected architectural components. A summary of the key approach to designing a suitable hybrid cloud architecture in this paper is to:

- Review state-of-the-art in hybrid cloud architecture with a focus on the non-functional attributes of availability, reliability, response time and throughput.
- Identify the main business requirements (hospital processes) to be executed in the cloud, classify them as workloads and determine how to measure their non-functional requirements also known as quality attributes.
- Determine the architectural components that are needed to make the data in the processes highly available to both internal and external users of the information system.
- Verify that the designed hybrid cloud architecture satisfies the specified non-functional requirements categorised under availability, reliability, response-time and throughput.

The rest of the paper is structured as follows: Section 2 presents the state-of-the-art in hybrid cloud architecture describing work that has been done to improve the performance of hybrid clouds. Section 3 describes the functional and non-functional requirements for the selected case, laying out the criteria for selecting a cloud computing architecture. Section 4 discusses the results of the application of the selected architecture and Section 5 concludes the article.

2 State-of-the-art in hybrid cloud architecture

2.1 Hybrid Cloud Architecture

The hybrid cloud model combines the benefits of computing resource elasticity in the public cloud whilst maintaining control of their sensitive data and mission-critical applications in a private on-premise infrastructure [27–30]. The performance of the cloud combination is an interesting subject of on-going study by many researchers as well as proposed solutions by providers such as VMware, IBM and Microsoft [31–33]. There are a few studies that have yielded strategies and techniques to help improve the overall performance and manageability of hybrid cloud facilities: proactive workload management is a technique treated by Zhang et. al. with the presentation of network architecture that handles workload spikes in the on-premise network by directing the excess into a shared or public network in the hybrid cloud [34]. Avresky et. al. proposed a framework that used machine learning to manage computing resources especially when IaaS deployed software anomalies are detected [35]. The strategy of the framework they proposed was to continue to receive and redirect virtual workloads to other geographic areas for processing even when deployment anomalies are detected. The afore-going proposed solutions are however not ideal for data-intensive applications in geographic locations where bandwidth costs are relatively high, such in sub-Saharan Africa [12].

Again, software and networking solutions have been put forth by cloud computing providers to improve the overall performance and manageability of the hybrid cloud, particularly to seamlessly bridge on-premise data centres with the public cloud infrastructure. Rackspace's offering of Microsoft Azure [32], IBM's Power8 architecture [31], VMware's Cloud Computing Platform [33] and Amazon's Hybrid Hosting package are some of the competing solutions targeted at large enterprises and SaaS providers however an analysis of the business processes and workloads are needed to determine their suitability in meeting specific business requirements such as in healthcare provider in need of a cold archiving solution [36]. Further, frequent workload transfer between data centres is apt to cost more, both financially and in quality of service especially at regions where the cost of Internet bandwidth is yet reach globally competitive prices [37].

A study that incorporated information processing operations into the design of cloud computing network was done by Haung et. al [38]. They reasoned that analysing the logs of the information systems could offer insights into how to maintain the working efficiency of the cloud platform in general. It however did not address the specificity of the hybrid cloud consisting of an on-premise data centre tethered to a public cloud. Again, a cloud network topology [39] that is able to handle a few hundred ecommerce order entries per minute will not be suited for transactional processing of big data in the order of tens of thousands per second. In order to benefit from a hybrid cloud setup that incorporates the above strategies and techniques, a business must consider its own business processes and constraints laid on its network architecture.

Figure 1 shows a general architecture underlying various adoptions of the hybrid cloud model. The hybrid cloud will increasingly be a preferred deployment model in enterprises for the long term primarily due to the need to support various business operations such as legacy applications running from their already established datacentres [26–28].

Migrating to the cloud takes careful planning and strategy to avoid extensive downtime in production applications or interruptions in services to clients. The cautious approach to public cloud adoption also reflects the need to remain compliant with regulatory provisions such as the HIPAA [40], the HITECH [41] and Sarbanes-Oxley Act [42]. Bandwidth costs are also factors of consideration especially where software applications are data intensive or involve the routine time-sensitive storage and retrieval of large files such as high resolution images or other multimedia files.

Major providers of cloud computing services offer solutions which guarantee compliance with regulations and some level of interoperability between the on-premise data centre and the public cloud, the latter functioning just as an extension of the former. These hardware and software solutions are configurable

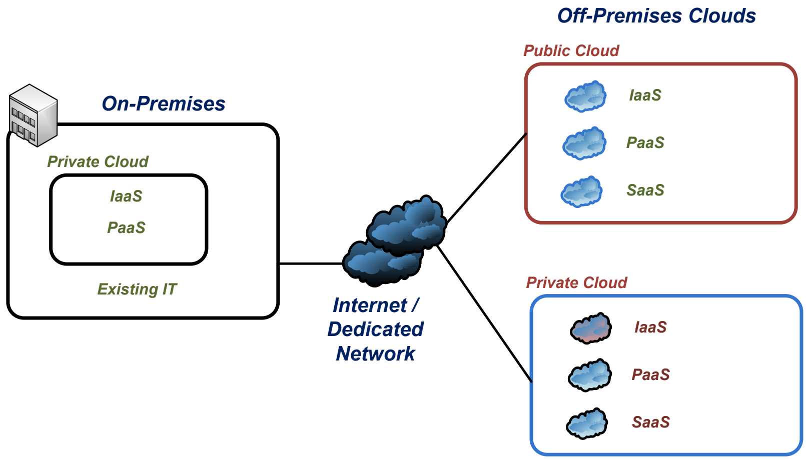


Fig. 1: Hybrid Cloud Architecture, [27]

to meet the data governance needs of businesses and help to route data to preferred storage locations within the hybrid cloud infrastructure. It can be said therefore that a hybrid cloud enables a business to maintain agility through public infrastructure provisioning and enjoy high utilisation in its on-premise facility [31–33].

Table 1: Layers In Cloud Architecture [43]

Layer	**Function**
Clients	End user devices
Services	XaaS: SaaS, PaaS, IaaS
Applications	
Platform	
Storage	Datacentre network fabric, computing and storage
Infrastructure	

Cloud computing architecture is often inspired by virtualisation and layerisation of its components that introduce flexibility in deployment to achieve intended purpose. Table 1 shows the basic layered architecture reported as consisting of clients, services, applications, platform, storage and infrastructure: where clients are the access devices; services, applications and platform represent the rendering of computing resources; and storage and infrastructure layers deliver the virtualisation environment [43]. For a hybrid cloud, the architecture is designed using a mix of components within layers, shown in Figure 2 and optimised to suit business goals. A hybrid cloud architecture will generally support business goals if it has a design that meets both the functional and non-functional requirements of the organisation's operations [46].

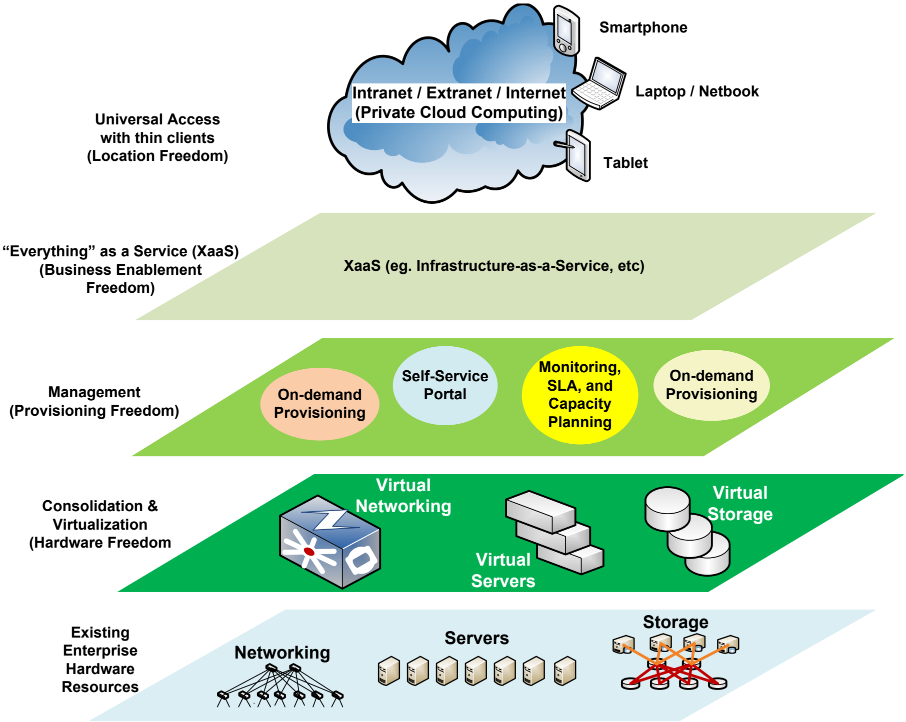


Fig. 2: Service-Oriented Infrastructure Framework, [44]

Figure 2 shows the location of the layers and the description of the relationships between them constituting the functional aspect of the network [46]. The functional view of the network architecture consists of the infrastructure, middleware and software that provide the basis upon which the components are drawn [44]. In deciding what architectural design to adopt, organisations and prospective providers usually base initial discussions on functional views of the required infrastructure.

The view as diagrams facilitates the discussion of concepts such as redundancy, replication, virtualisation and load balancing, all essential elements in making a cloud computing architecture more resilient. The functional view of cloud computing architecture is commonly represented using components grouped by functionality.

The cloud architecture likened to a giant computer is represented with groupings of components as shown in Figure 3. The server module can be thought of as the "brains" or processor of the computer [44]. It consists of virtual machines consolidated onto one or more physical computers. They are interconnected using the internal datacentre network and managed through virtualisation technologies such as the virtual machine monitor or hypervisor.

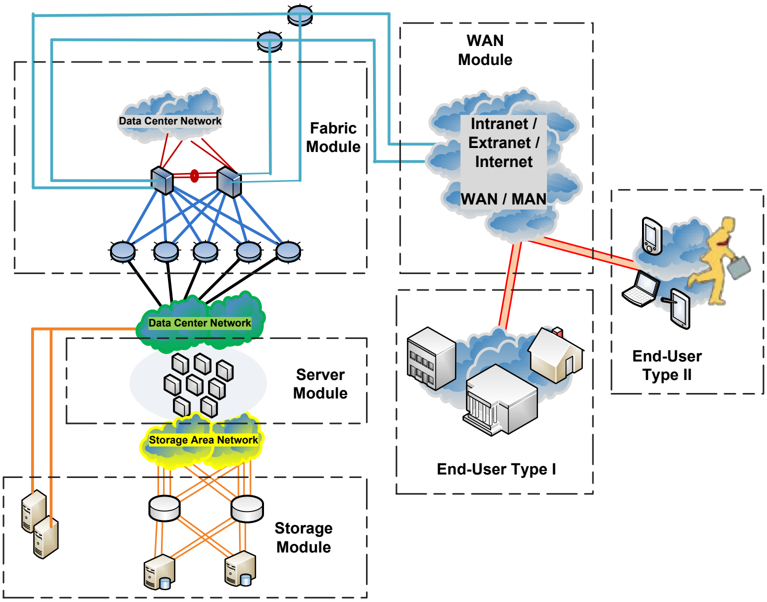


Fig. 3: Modular view of cloud computing architecture, [44]

The storage module consists of hard disks arranged in various configurations such as in arrays and using technologies such as RAID to provide redundancy and resilience. Even though the storage module is connected to the server module via the storage-area network, it also connects directly to the internal datacentre network using fast links such as Fibre Channel over Ethernet (FCoE). The fabric module is the core network that integrates the FCoE and other ethernet connections and technologies including facilitating connections outside the datacentre. The WAN module can either be the intranet, extranet or Internet, facilitating connectivity of end-users to the datacentre. Depending of the type of connection, wired or wireless, the end-users can be grouped into the ones in fixed locations and mobile users respectively.

The non-functional view of the network consists of the architecturally significant requirements that must be considered in implementing a hybrid cloud. Because the hybrid cloud architecture facilitates the gradual adoption of cloud computing, constraints are placed on the architecture to ensure there is minimal disruption to existing essential IT services and also to specify non-negotiable requirements in the new network. The non-functional view highlights the architecturally significant requirements affecting the behaviour of the hybrid cloud in supporting specific business goals. For instance in building a hybrid cloud for high availability of data, non-functional specifications to generally consider are capacity and throughput, bandwidth utilisation, offered load, accuracy, efficiency, latency, response and device CPU utilisation. Not all non-functional requirements directly affect the architecture and indicators include requirements that are strict, constraining, limiting or non-negotiable such as requirements associated with high biz value; requirements of high importance to stakeholders; requirements not addressed by existing components; QoS requirements; and requirements with potential for causing budget overruns.

Workload can be defined as a service or collection of code that can be executed or the amount of work that needs to be accomplished by computer resources in a certain amount of time [46]. There are five patterns of workload in cloud computing (shown in Table 2), each focused on achieving specific goals.

Table 2: Workload Types and their Focus [45, 48]

Workload Type	**Workload Pattern**	**Description**	**Focus or Goal**	**Time Sensitivity**
Batch workload	Periodic	Designed to operate in background eg. logs	Process large volumes of data in the background	Not time sensitive
Transact-ional workload	Unpredictable, Continuously Changing	Automation of business processes such as billing/order processing. If very complex, it is best to let it stay on-premises	Focuses on large volumes of current transactions	Typically requires real-time processing
High performance workload	Unpredictable, Continuously Changing	Used for scientific/ technical and often complex operation. Environment usually must be optimised for them	Has scientific or technical focus	Requires high amounts of compute resources for normal processing
Analytic workload	Periodic, Static	Typically make sense of vast amounts of data across a complex hybrid environment in real-time	Affects large amounts of data for decision making	Depending on the business it could either be batch or real-time
Database workload	Continuously Changing	The most common type of workload. It must be tuned and managed to support the service using the data.	This is highly tuned to application needs	May require specialised hardware integration

IaaS	PaaS	SaaS
Business Process	Business Process	Business Process
Applications	Applications	Applications
Data	Data	Data
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
O/S	O/S	O/S
Virtualization	Virtualization	Virtualization
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

Fig. 4: Service Responsibility Line, [27]

Depending on the type of cloud deployment model and architecture, workloads may be achieved the required performance. Organisations could have a variety of workloads in their infrastructure which can benefit from a combination of cloud deployment models to save cost. The hybrid cloud can also provide resiliency to overall application availability and faster processing. It thus calls for well-architected and abstracted workloads consisting of multiple services in multiple locations.

The economics of cloud computing are very much affected by the workload requirements. Transactional workloads such as email, collaboration and messaging are well-suited for the public cloud due to features like standardisation, optimisation and scalability. Specialised workloads such the quarterly running of financial reports, a private datacentre is the most appropriate as it is likely the organisation has already invested in the datacentre and hence no special cost-savings moving the workload to the cloud.

Economic benefit can be delivered from the public cloud if there is a need for increased capacity for seasonal handling of workloads, or for software evaluation or system testing. Under these circumstances it will not be economically sound to just build infrastructure due to the short time duration [49]. It is simply better to be up and running within the shortest time and scale-in when the extra capacity is no longer needed. The economic impact and responsibility of running the cloud infrastructure also depends on the service model chosen to handle identified workloads. The service responsibility line in Figure 4 shows the level of economic investment and responsibility required for each cloud service model.

2.2 Specific Requirements

The business requirements of the hybrid cloud are categorised into the standard engineering requirements: network, archiving, backup and recovery, and compliance and regulatory requirements [27]. The network requirement states the hybrid cloud shall have access and offloading capabilities for mobile devices, wired connection of desktop computers and fibre and radio for WAN interconnection. The compliance and regulatory requirements states the hybrid cloud shall have all personally identifiable information (PII) stored in the private datacentre in compliance with the Data Protection Act 2012 of Ghana [54]. This applies to both data at rest and data in transit. The archiving requirement states the hybrid cloud shall have the capacity to archive inactive records for a specified period of five years. The record in the archive shall be fully retrievable within 30 minutes of initiating request. The backup and recovery requirement states the hybrid cloud shall have the capacity to make backup once a week and the system should be fully restored within four hours of the initiating the recovery process. On the mobile access and offloading requirement, the network architecture is further enhanced to support mobile computing where mobile devices offload resource intensive tasks to the hybrid cloud. The hybrid cloud is thus configured to provision resources in the public cloud whenever workloads in the datacentre exceed a maximum threshold. The functional requirements culminate in a hybrid cloud design whose suitability to meet the peculiar requirements of a healthcare computer network is determined by conducting metric measurements involving availability, reliability, response time and throughput [25]. To serve as a reference for the rest of the paper, metric as a term and the quality attributes of the hybrid cloud that is to be measure are introduced.

Metric is defined by NIST as “a standard of measurement that defines the conditions and the rules for performing the measurement and for understanding the results of a measurement” [50]. Availability refers to the capability for introduced redundancies to mask errors and failures that occur in the hybrid cloud system and ensure continuous running of processes [25]. Reliability “Refers to the ability to ensure a continuous process of the program without loss.” It is a measure of how reliably a system can recover after failure. The reliability metric category has as important measures the Mean Time To Recovery (MTTR) – how long it takes for a system to recover from failure, Mean Time Between Failures (MTBF) – amount of time that elaspsed between failures and Recovery Time Objective (RTO) – determines how long the entire system is down [25]. Response Time: “This is defined as the time it takes for any workload to place a request for work on the virtual environment and for the virtual environment to complete the request” [51]. Other synonyms for this metric are agility and adaptability [25]. Response time has a direct impact on application performance and availability in the cloud. Throughput “refers to the performance of tasks by a computing service or device over a specific period” [51]. The metric category is used for measuring rate of transactions as well as the rate of data transported – in bits per second. The workload defined as the amount of work that needs to be accomplished by computer resources in a certain amount of time [45] is used as the basis measuring the metrics.

In the next section, the high level requirements are discussed in terms of the preceding networking requirements.

3 The Case of the Selected Hospital

This section describes the functional and non-functional requirements of a hybrid cloud infrastructure that can support data access and clinical operations of the case study. The process model (flow of data) of the hospital and user characteristics is first presented to give an overview of the services that depend on the cloud infrastructure. The networking requirements that best support the process model are then determined with discussion of the various components. The areas of networking requirements to be discussed are availability, economic archive storage capacity, mobile device accessibility, regulatory compliance and security.

3.1 The Hospital

The selected case in Ghana has nine facilities geographically spread across a city. The clinics function as centres of primary healthcare and the hospitals serve as referral facilities if further treatment is required. Patient records are accessible from any facility location by authorised hospital and clinic staff. The records are typically composed of bio-data, diagnoses information, laboratory and radiology investigation results, prescriptions and billing information. The radiology investigations result in the production of high resolution medical images that need to be stored as part of the patient history. High resolution videos stored on the local network aside the patient records and used to facilitate in-house training of staff. Some other workloads cover administrative processes of the hospital such as human resources, accounting, maintenance management and internal VoIP communication. The network is accessible by external users for rendering various complementary services. In situations of referral of patients to any of the neighbouring national and regional hospitals, the external specialist is able to access the hospital's corporate network to retrieve the patients' medical history especially in an emergency where the physical folders are not immediately available. Medical insurance companies access a read-only version to vet the medical bills of patients who have policies with them. The general idea is to grant them minimal access to verify prescriptions and other treatment on the basis of the diagnoses given; and raise queries on any billing information that raises doubts about policy compliance. The ambulatory services whilst enroute to the hospital with a sick patient access their electronic records to improve emergency treatment. Access to the data is by a mix of desktop computers and mobile devices typically smartphones and tablets. The performance of the hospital information system in handling such workloads depends on the server workload and bandwidth especially for access outside of the corporate network. The hospital information system is hosted on datacentre servers located in one of the Hospital buildings. The data centre consists of two rack-mount servers having a total RAM of 64GB, 2TB of hard drive space with an additional Network Attached Storage (NAS), and multi-core processors. A hypervisor installed manages four virtual servers that handle the hospital information system, HR, accounting information and database management systems, VoIP gateway services and network management tools. Laboratory and radiology equipment connect directly to the hospital information system via a multi-layer switch in a machine-to-machine communication.

High availability as a network requirement is embodied in the redundancies of the services and their provisioning from public providers with self-healing infrastructure [52, 53]. Availability ensures access to the patient and administrative data at all times taking into consideration the unpredictable nature of the public cloud access. Bandwidth remains a critical factor in connecting two or more clouds together. Significant data movement between the public cloud and the on-premise data centre constrains the corporate internet in terms of cost and bandwidth with the latter resulting in increased latency from transmission queues on the public cloud and overall unpredictable quality of service. The design of the hybrid cloud infrastructure must thus ensure minimal intervening equipment in the path of data packets to generally minimise the transmission times of its relaying or forwarding. On the software side, dynamic caching, compression and pre-fetching are some of the web acceleration technologies that help improve end user connectivity. If an application however is chatty and requires loads of data transfer across clouds, then adopting cloud computing generally becomes a difficult decision.

Storage as a requirement facilitates the archiving of patient health records in the hybrid cloud design. The physical location of storage in a hybrid cloud is an essential factor in determining the overall cost owning and maintaining the data. Though new technologies enable storage of more volumes of data at lower costs, the velocity of generating new data continues to rise with the prolific integration of IoT into business operations. Thus the preparation of the data centre for IoT will need high performance redundant connectivity with the LAN, making use of Fibre Channel or Fibre Channel over Ethernet as connections between the servers and SAN with the datacentre. In designing a hybrid cloud infrastructure, the possibility of having a backup and disaster recovery setup without the typical associated upfront costs is an advantage and a quick, reliable data backup and restoration plan lies at the heart of system availability. The cloud offers elastic resource provision that leaves hospital administrators to focus on the core business of providing healthcare. On which side of the hybrid cloud to store the information system data is one decision that impacts the overall performance of the system and is constrained by data governance policies of the organisation.

Regulation in Ghana [54]] also enjoins healthcare institutions to keep archived electronic patient records for a minimum period of five years. This requires an archiving of old data to reduce the operational load on the central databases. Depending on the frequency of update of the archival data, active, cool and cold archiving may be chosen. The colder the archive the slower the retrieval and the less expensive it is. Block storage on the other hand it is fast to access with low latency but also more expensive per megabyte of storage space and bandwidth. One other inexpensive option to archiving is to use the tape though it can be slow in both saving and retrieval of data.

Security was the next priority in the design that identified each user in the network for appropriate and authorised access whilst guarding against the loss and alteration of data. The measures of security: confidentiality, integrity, authenticity and availability were to be factored into the design using mechanisms and controls to safeguard or improve the security. The level of insecurity was to be measured using vulnerabilities, threats and risks of migrating the existing system to the hybrid cloud. Again the new network had to be evaluated for possible attacks from hidden threat agents such as the anonymous attacker, malicious service agent, the trusted attacker and the malicious insider. Security embodies the mitigation of possible threats and vulnerabilities that can arise if safeguards are not put in place. Compliance deals with the application of regulatory policies to electronic data transactions within the information system. For the hospital, health data was to be held private and confidential and especially ensuring personally identifiable information are protected.

The hybrid cloud introduces complexity into the corporate network and departs from the traditional singular application tools with which IT staff are familiar. The public cloud has a different set of management tools for monitoring, provisioning and decommissioning. It is highly desirable to reduce the cost of administration and increase staff efficiency by having one application that administrators can use but there are few management tools that can efficiently and effectively administer both the public and the private cloud.

Finally, to support mobile devices, the mobile cloud computing paradigm has to be factored into the overall design of the hybrid cloud architecture. The mobile devices offload heavy tasks into the cloud system in order to conserve its local resources especially its limited battery life [7, 8]. The connection to the cloud system is typically via Wi-Fi however other connection methods such as Bluetooth and the regular cellular data packages.

3.2 The Requirements for a Hybrid Cloud in the Hospital

For the selected case of the hospital, metric values are provided as thresholds for acceptable performance and to support mobile access. It is required that the hybrid cloud have a link availability of 99.90% which translates to 526 minutes a year in downtime. Availability in the case of the Hospital is defined based on the weekly user experience of clinical staff interacting with the hospital information system and with a maximum allowable downtime of 10.11 minutes per week. This in effect implies the information system should be available for access 99.9% of all the time within the year and any cumulative downtime of about 526 minutes or more means the systems is below the required availability threshold. The hybrid cloud introduces more redundancies both at the network level (LAN and WAN) and the application level (Application server, Web server and Database server) to ensure services are available when needed.

Reliability in the hybrid cloud refers to the "engineered availability" [47] of the components and setup in terms of not failing for a period of time: Mean Time To Failure (MTTF); not failing frequently: Mean Time Between Failures (MTBF) and how long it takes to recover from failure: Mean Time To Recovery/Repair (MTTR). The "measured availability" is the actual measurement performed on the engineered setup and components: calculated as MTBF/(MTBF + MTTR) [47] and is required to have a value not less than 99.5% in the hospital network.

The response time is taken from the perspective of the application response time (ART) which is the sum of the network response time (NRT) and the transaction response time (TRT). By separating the time elapsed due to the network from that generated by the application itself, it is possible to determine the response time of the hybrid cloud network alone. Baseline measurements performed during peak and off-peak times in the WAN portion of the hybrid cloud facilitate more stable values due to greater control over the network components in the WAN and datacentre. Throughput is measured as the quantity of data (bits) transmitted in one second between the hospital applications server and the client computer. The requirement in the hospital is to have the datacentre network deliver at close to gigabit LAN rates of 10 Mbps at non-busy periods and 5 Mbps at peak usage for over 500 users.

3.3 On-site Requirements Information

The connection of the hospital WAN to the public cloud resulted in a hybrid cloud shown in Figure 5 through which each hospital branch accesses both the internet and the intranet. The VPN through the internet has the datacentre dedicated bandwidth of 10Mbps whilst the branches of the hospitals had dedicated bandwidths of 2Mbps. During measurement of the selected metrics the Wireshark network protocol analyser was employed in measuring the throughput and Telerik Fiddler a web debugging tool was used to measure the response times of the network. During the two month period of measurement, the network experienced downtimes on average every fourth day for an average period of 4minutes giving a measured availability of (5760 minutes / (5760 + 4 mins)) = 99.9306%. Figure 6 shows the average response time experienced in running the hospital applications: 1,092,692 bytes transferred from server in 4.432 - 3.818 = 614 milliseconds giving a throughput of about 1.69MBytes per second. An average of 1.35Mbytes per second was experienced transferring small radiology images from the server to a client computer.

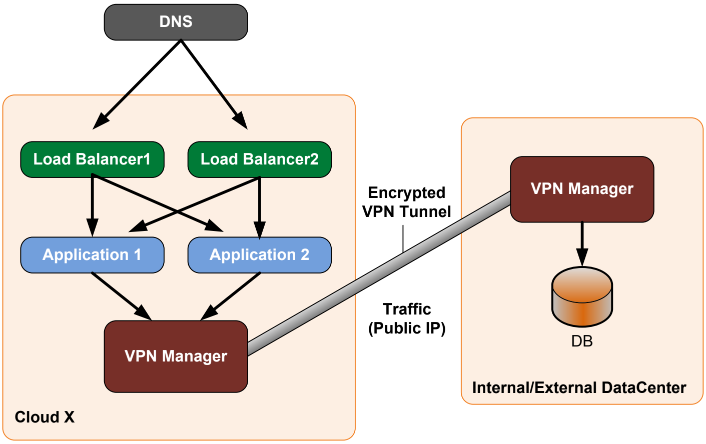


Fig. 5: Hybrid Cloud to enhance availability and ensure regulatory compliance, [55]

4 Results and Discussion of Hybrid Cloud Design

This section presents the architectural design resulting from the consideration of requirements that will support the peculiar business goals of the case study. The objectives of the architecture were to introduce elements of design that will give high availability, storage, mobile access, compliance and security to clinical data and operations using the hybrid cloud.

The measured quality metrics were well within the requirements validating the introduction hybrid cloud computing into the hospital network. To further ensure availability of services on the hybrid cloud, redundancy was built into the overall architecture, introducing an extra load balancer in the network, multiple servers in an array and database replication apart from the snapshot backups taken at regular intervals. Accessing and processing of healthcare data on mobile devices was facilitated by the server array which allows the spawning of application instances to handle tasks offloaded from resource poor mobile devices. The extra load balancer ensured that incoming requests for processing healthcare data are handled by a highly available server within the datacentre or requests are redirected to the public cloud when fixed resources are low on capacity.

Storage requirements were fulfilled with a combination of replication in the master database in the on-premise datacentre and a pay-per-use public cloud storage. The pay-per-use storage in the public cloud was introduced to cost-effectively archive old patient records and clinical history such that retrieval was achieved in minimal times. Bandwidth costs were thus saved when cold data was pushed into the cloud and more active was retained on the data centre

```
Request Count:   1
Bytes Sent:      919      (headers:919; body:0)
Bytes Received:  1,092,981       (headers:289; body:1,092,692)

ACTUAL PERFORMANCE
-------------
Client Connected:    04:33:01.208
ClientBeginRequest:  04:34:02. 408
GotRequestHeaders:   04:34:02.408
ClientDoneRequest:   04:34:02.408
Determine Gateway:   0ms
DNS Lookup:          0ms
TCP/IP Connect:      0ms
HTTPS Handshake:     0ms
ServerConnected:     04:33:33.027
FiddlerBeginRequest: 04:34:02.408
ServerGotRequest:    04:34:02.412
ServerBeginResponse: 04:34:03.818
GotResponseHeader:   04:34:03.818
ServerDoneResponse:  04:34:04.432
ClientBeginResponse: 04:34:04.432
ClientDoneResponse:  04:34:04.432

       Overall Elapsed:     0:00:02.024

RESPONSE BYTES (by Content-Type)
-------------
text/html: 1,092,692
~headers~: 289
```

Fig. 6: Response times as measured with Fiddler

servers. The cost savings were further augmented with a VPN tunnel to meet the security and compliance requirements of the hybrid cloud infrastructure.

5 Conclusion

The paper looked into how to architect a hybrid cloud infrastructure to enhance information whilst support mobile device access. A case of a hospital in Ghana which had a unique need for hybrid cloud adoption was examined and the architectural redesign of their existing facilities was found to enhance support for clinical operations across all their facilities.

Hybrid cloud as growing trend is expected to shape the cloud computing landscape as the interoperability challenges associated with its adoption are addressed. Inroads on Software Defined Networks (SDN) and Software Defined Storage (SDS) will further culminate in smoother hybrid cloud management and easier aggregation of control onto a single pane of glass. More and more software would be born cloud-ready to scale horizontally on demand and interoperate with other software via microservices architecture. With adequate mobile device centred planning of hybrid cloud computing infrastructure, more organisations in the developing world can improve data sharing among its key stakeholders and the increasingly mobile-savvy clientele. Leveraging on cloud computing technology and the high mobile phone penetration rates is an interesting way to transform existing corporate data centres into data sharing platforms for socio-economic growth, especially in sub-Saharan African economies.

References

[1] P. Mell and T. Grance. The NIST definition of cloud computing. *NIST Special Publication*, 145:7, 2011.

[2] Cloud Management Software | RightScale. *http://www.rightscale.com/products-and-services/products/cloud-management.*

[3] Gartner Says Worldwide Public Cloud Services Market to Grow 17 Percent in 2016. *http://www.gartner.com/newsroom/id/3443517.* [Accessed: 31-Dec-2016].

[4] GSMA. GSMA Mobile Economy 2016. *http://www.gsma.com/mobileeconomy/*

[5] Kelly Hill. Cloud computing emerging in Africa - RCR Wireless News 2015. *http://www.rcrwireless.com/20151023/featured/ cloud-computing-in-africa-tag6*

[6] Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, 2016.

[7] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications. *IEEE Wireless Communications*, 2013.

[8] S. Qureshi, T. Ahmad, K. Rafique, T. Shuja-ul-islam, and K. Rafique. Mobile cloud computing as future for mobile applications - Implementation methods and challenging issues. in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 2011:467—471.

[9] N. Fernando, S. W. Loke, and W. Rahayu. Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84—106, 2013.

[10] P. Nawrocki and W. Reszelewski. Resource usage optimization in Mobile Cloud Computing. *Computing Communications*, 2016.

[11] J. Hanen, Z. Kechaou, and M. Ben Ayed. An enhanced healthcare system in mobile cloud computing environment. *Vietnam Journal of Computer Science*, 3(4):267-–277, 2016.

[12] A. Neyem, M. J. Carrillo, C. Jerez, G. Valenzuela, N. Risso, J. I. Benedetto, and J. S. Rojas-Riethmuller. Improving Healthcare Team Collaboration in Hospital Transfers through Cloud-Based Mobile Systems. *Mobile Information Systems*, 2016:1—14, 2016.

[13] H. Wu, Q. Wang, and K. Wolter. Mobile Healthcare Systems with Multi-cloud Offloading. in *2013 IEEE 14th International Conference on Mobile Data Management*, 2013:188-–193.

[14] N. Y. Asabere. mMES: A Mobile Medical Expert System for Health Institutions in Ghana. *International Journal of Science and Technology*, 2(6):333–344, 2012.

[15] M. Dumiak. E-health's future frontiers. *Bulletin of the World Health Organisation*, 90(5):328-–9, 2012.

[16] M. Yuvaraj. Determining factors for the adoption of cloud computing in developing countries. *Bottom Line*, 29(4):259-–272, 2016.

[17] S. K. Sharma, A. H. Al-Badi, S. M. Govindaluri, M. H. Al-Kharusi, and M. H. Al Kharusi. Predicting motivators of cloud computing adoption: A developing country perspective. *Computers in Human Behaviour*, 62:61—69, 2016.

[18] A. A. Dahiru and A. A. Dahiru. Cloud Computing: Adoption Issues for Sub-Saharan Africa SMEs. *Electronic Journal on Information Systems in Developing Countries*, 62, 2014.

[19] H. Gangwar, H. Date, R. Ramaswamy, H. Date, and R. Ramaswamy. Developing a Cloud-Computing Adoption Framework. *Glob. Bus. Rev.* 16(4):632-–651, Aug. 2015.

[20] Y. Hu, F. Lu, I. Khan, and G. Bai. A Cloud Computing Solution for Sharing Healthcare Information. *The 7th International Conference for Internet Technology and Secured Transactions*, 465–470, 2012.

[21] Y.H. Kuo, Y.L. Jeng, and J.N. Chen. A Hybrid Cloud Storage Architecture for Service Operational High Availability. *IEEE 37th Annual Computer Software and Applications Conference Workshops*, 487-–492, 2013.

[22] L. Marshall. Weigh total costs moving to the cloud: a hybrid model offers the best of both worlds. *Manag. Healthc. Exec., vol. 22, no. 2*, p. 30-, 2012.

[23] N. Regola and N. V Chawla. Storing and Using Health Data in a Virtual Private Cloud. *J. Med Internet Res, vol. 15, no. 3*, 2013.

[24] G. Gavrilov and V. Trajkovik. Security and Privacy Issues and Requirements for Healthcare Cloud Computing. *ICT Innovations 2012 Web Proceedings*, 143–152, Oct. 2012.

[25] A. K. Bardsiri and S. M. Hashemi. QoS Metrics for Cloud Computing Services Evaluation. *Int. J. Intell. Syst. Appl., vol. 6, no. 12*, pp. 27–33, Nov. 2014.

[26] L. Griebel, H.-U. Prokosch, F. Köpcke, D. Toddenroth, J. Christoph, I. Leb, I. Engel, and M. Sedlmayr. A scoping review of cloud computing in healthcare. *BMC Med. Inform. Decis. Mak.*,15(1):17, Dec. 2015.

[27] Cloud Standards Customer Council. *Practical Guide to Hybrid Cloud Computing*, 29–31 2016.

[28] R. Hill, L. Hirsch, P. Lake, and S. Moshiri. *Guide to Cloud Computing*, Springer London, 2013.

[29] M. Waschke. *How clouds hold IT together: integrating architecture with cloud deployment*, 7–8 Apress, 2015.

[30] S. Srinivasan. *Cloud Computing Basics*, 112–113, Springer New York, 2014.

[31] The Hybrid Cloud Built to Perform with POWER8 in IBM SoftLayer, 2016. *https://www-304.ibm.com/events/tools/interconnect/2016ems/REST/presentations/PDF/InterConnect2016_2751.pdf.* [Accessed: 14-Nov-2016].

[32] The Hybrid Cloud and Microsoft Azure, 2015. *http://docs.media.bitpipe.com/io_12x/io_122367/item_1162106/RS_HybridAzure whitepaper_v2.pdf.* [Accessed: 14-Nov-2016].

[33] Cloud Management Platform: VMware. *http://www.vmware.com/ solutions/virtualization/ cloud-management.html.* [Accessed: 13-Nov-2016].

[34] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen. Proactive workload management in hybrid cloud computing. *IEEE Trans. Netw. Serv. Manag.*, 11(1):90-–100, Mar. 2014.

[35] D. R. Avresky, P. Di Sanzo, A. Pellegrini, B. Ciciani, and L. Forte. Proactive scalability and management of resources in hybrid clouds via machine learning in Proceedings. *2015 IEEE 14th International Symposium on Network Computing and Applications* 114-–119, 2015.

[36] M. S. Maaref. Cloud computing in Africa - Situation and perspectives. *Telecommun. Dev. Sect.- ITU*, p. 70, 2012.

[37] Numbeo. Price Rankings by Country of Internet (10 Mbps, Unlimited Data, Cable/ADSL) *https://www.numbeo.com/cost-of-living/country_price_rankings?itemId=33*

[38] K. Z. Haung, Y. S. Chen, and Y. K. Chung. Perspectives on process mining within cloud computing. *3rd International Conference on Advanced Computer Control, ICACC 2011*, 2011, pp. 656–660.

[39] M. Tsugawa, A. Matsunaga, and J. Fortes. Cloud Networking to Support Data Intensive Applications. *Cloud Computing for Data-Intensive Applications*, New York, USA, 2014.

[40] P. Law. Health Insurance Portability and Accountability Act of 1996. Public Law 104-191, United States statutes at large, 1996. *https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996.*

[41] HITECH Act Rulemaking and Implementation Update. *http://www.hhs.gov/hipaa/for-professionals/security/guidance/HITECH-act-rulemakingimplementation-update/index.html*[Accessed: 03-Mar-2017].

[42] The Sarbanes-Oxley Act 2002. http://www.soxlaw.com/[Accessed: 03-Mar-2017].

[43] A. Kalapatapu, M. Sarkar, and H. Khazaei. Cloud Computing. *Cloud Computing*, CRC Press, 2011, pp. 3–29.

[44] S. R. Smoot, N.-K. Tan, and N.-K. Tan. *Private cloud computing: consolidation, virtualization, and service-oriented infrastructure*, Morgan Kaufmann, 2012.

[45] F. Halper, M. Kaufman, and J. Hurwitz. *Hybrid Cloud For Dummies*, Wiley, 2012.

[46] J. D. McCabe. *Network analysis, architecture, and design*, Burlington: Elsevier/Morgan Kaufmann Publishers, 2007.

[47] Availability Measurement. *http://www.cisco.com/networkers/nw04/presos/docs /NMS-2201.pdf.* [Accessed: 03-Mar-2017].

[48] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter. *Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications*, 2014.

[49] G. Laatikainen, O. Mazhelis, and P. Tyrvainen. Cost benefits of flexible hybrid cloud storage: Mitigating volume variation with shorter acquisition cycle. *J. Syst. Softw.*, vol. 122, pp. 180–201, Dec. 2016.

[50] NIST Cloud Service Metrics Sub Group, *Cloud Computing Service Metrics Description*, 2015.

[51] Cloud Services Industry's 10 Most Critical Metrics - Guiding Metrics. *http://guidingmetrics.com/content/cloud-services-industrys-10-most-critical-metrics/.* [Accessed: 20-Feb-2017].

[52] Y. Dai, Y. Xiang, and G. Zhang. Self-healing and Hybrid Diagnosis. in *Cloud Computing*, Springer Berlin Heidelberg, 45-–56, 2009.

[53] Do you have a self-healing cloud? Cloud computing news. *https://www.ibm.com/blogs/cloud-computing/2012/03/do-you-have-a-self-healing-cloud/.*

[54] Data Protection Commission: protecting the privacy of the individual & personal data. *https://www.dataprotection.org.gh/.*

[55] RightScale. Cloud Computing System Architecture Diagrams. *http://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.html.*

Paper B

Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis

Azumah, Kenneth Kwame, Tadayoni, Reza and Sørensen, Lene Tolstrup

The paper has been published in the
IEEE 7th International Conference on Adaptive Science & Technology (ICAST), pp. 1–8, 2018.

The layout has been revised.

Abstract

Hybrid cloud computing enjoys the benefits from the two worlds of private cloud and public cloud. The combination of the two unique clouds introduces new challenges that must necessarily be addressed before adoption. Much work has been done tackling these challenges from various perspectives that have contributed to a diffusion of the main issues confronting the hybrid cloud. A harmonization of the various research themes was the motivation for this study. The goal was to identify, make a high-level categorization and describe thematic areas that address the unique challenges arising from combining two or more distinctive cloud deployment models. This paper, through a qualitative systematic literature review contributes to the synthesis of themes arising from proposed interventions to hybrid cloud challenges. A total of 34 studies were synthesized resulting in three thematic areas of data security, cost optimization and performance enhancement. The study showed majority of research on hybrid cloud challenges belonged to one or more of these thematic areas that can aid enterprise leaders formulate strategies for hybrid cloud adoption.

Keywords: Hybrid cloud, data security, cost optimization, meta-analysis

1 Introduction

Hybrid cloud is seen as solution to computing challenges that cannot be solved by the purely public or private cloud. The elasticity of the one or more public cloud resources combined with the maximum control in the private on-premise infrastructure helps to overcome the issues confronting the separate deployment models. This combination is not however without its own challenges birthed from the need to make trade-offs in balancing cost, performance and security [1]. Majority of studies on the hybrid cloud focus on solving the challenges arising from the complexity of operating across the two or more individual cloud deployments and this has resulted in a large body of highly varied literature proposing equally varied strategies.

This paper is a systematic review of studies that focused on the optimization of cost, performance and data security in the hybrid cloud. The research question is what are the strategies being adopted to improve performance and reduce cost of operating the hybrid cloud? The relevance of the research question is seen in the trade-offs that have to be made between data security and performance on one hand, and cost and performance on the other. The nature of the challenges confronting hybrid cloud adoption militates against the simultaneous improvement on all three fronts resulting in a plethora of studies proposing a wide range of strategies and solutions. This paper attempted to find the common themes emerging from literature through a qualitative systematic review and a comprehensive analysis of the strategies and solutions emerging thereof. The next section 2 provides a background on the characteristics of the hybrid cloud and dwells on three key areas of academic research (sections 2.1, 2.2 and 2.3). Related studies are next presented in section 2.4 providing their areas of focus and gaps identified. Section 3 presents the method and steps followed in conducting the systematic literature review and section 4 presents the results of the analysis. Discussions are carried out in section 5 laying out hybrid cloud thematic areas discovered in the systematic review and followed by the conclusion in section 6.

2 Background

The concept of hybrid cloud has generated some level of confusion owing to a generally loose definition that puts together two or more cloud deployment models. A deployment model refers to how the cloud infrastructure is organized: either as a public, private or community cloud. An article [2] commented on a working definition of the hybrid cloud given by Forrester Research[1] as "One or more public clouds connected to something in my data center. That thing could be a private cloud, that thing could just be traditional data center infrastructure." Hybrid cloud combinations typically are on/off-premise private cloud and one or more public clouds [3]. This is not be confused with other cloud combinations such as the multicloud which represents the situation of a customer using multiple public cloud providers only, or intercloud which is a concept on the pooling of public clouds for economic purposes [4].

The motivating factors for hybrid cloud adoption include making up-front cost-savings on computing infrastructure, securing sensitive data from getting into the untrusted public cloud, increasing the performance of workloads when private cloud capacity is reached and serving as a backup solution typically for the private on-premise infrastructure. The main aim is to benefit from on-demand and elastic scalability of the public cloud whilst maintaining full control of sensitive data and mission-critical applications running in the private on-premise infrastructure. The benefits however go with the challenge of managing complexity and costs since not all hybrid cloud deployments necessarily culminate in cost savings. Facebook moved Instagram and WhatsApp workloads from Amazon's EC2[2] and IBM's SoftLayer[3] respectively to its own private datacenters to achieve cost savings because it had the economies of scale that beat the hybrid cloud deployment option [5].

2.1 The Economics of the Hybrid Cloud

Whether cost-savings are to be made on adopting a hybrid cloud and what strategies to pursue have been studied widely by cloud economists. Costs

[1] https://en.wikipedia.org/wiki/Forrester_Research

[2] http://www.datacenterdynamics.com/content-tracks/servers-storage/facebook-ditches-aws-to-bring-instagram-data-in-house/87619.fullarticle

[3] http://www.datacenterdynamics.com/content-tracks/colo-cloud/facebook-to-move-whatsapp-workloads-from-ibms-cloud-to-its-own-data-centers/98465.fullarticle

of running the hybrid cloud can be classified into quantitative costs which consists of service charges and usage patterns; and qualitative costs which are affected by human biases [5]. Quantitatively, mathematical models aid cloud cost calculators to make the process of determining cost of migration relatively easy. Major cloud providers including Microsoft, Amazon, IBM, Rackspace and Google have provided their calculators for the convenience of customers. However, [5] in proposing his mathematical model for costing hybrid cloud adoption cautions that additional costs such as cloud management and data transfer must be considered for completeness of the model.

A study by [6] that closely approached the costing challenge applied the total cost of ownership (TCO) theory in developing a mathematical model that supports decision making for cloud computing. The cost types identified included costs for strategizing service charges, maintenance and evaluation costs. The cost factors that influence the cost types included expenditure of time, consultancy and support costs, and costs of computing resources. Their model took into consideration the cost type and cost factors, and aggregated various combinations to cover all aspects of costs in the cloud adoption.

2.2 Performance Issues in the Hybrid Cloud

Performance aspects of the hybrid cloud focus on the response time of processing workloads at an optimal cost. More computing resources can be spawned in the public cloud to speed up processing however there are constraints of data communication costs especially for data-intensive applications and the constraint of keeping sensitive data in the private cloud [7]. Also business-critical applications that perhaps were developed in-house often are not candidates for migration into the public cloud and must be managed within the private side of the hybrid cloud. Performance also connotes reliability, robustness and efficiency of the hybrid cloud infrastructure. Scheduling algorithms that detect when to burst permissible workloads into the public network must be efficient for the planned architecture [8].

2.3 The Security Aspects of the Hybrid Cloud

Security remains one of the major challenges to address in hybrid cloud adoption. The challenge can be grouped into data and network security, regulatory compliance and network availability. The ease with which sensitive-data can leak to the public cloud requires extra preventive measures to be taken in a hybrid network. In this situation workloads have to be carefully scheduled to avoid exposing sensitive data into an untrusted network. Again there is the challenge of applying private cloud security policies in the public cloud. This is especially complex for identity access management where user will have access to both sensitive data and the public cloud introducing additional costs. Network related attacks can originate from the public cloud more easily due to the integration putting a previously isolated private cloud at risk in a hybrid cloud [9], [10].

2.4 Related Work

There have been diverse systematic reviews of cloud computing literature but few focused specifically on the hybrid cloud and its emerging issues. Ullah & Khan [1] in their systematic literature review identified key challenges that are to be addressed when adopting a hybrid cloud. Among them were public cloud security concern, effective management issue and integration complexity listed as the top three challenges that must tackled critically in hybrid cloud adoption. In dealing with these constraints militating against user adoption of the hybrid cloud, their study took cognizance of the strategies employed by researchers in arriving at solutions. Their study stopped short of analyzing the various strategies but found the experimental approach as the most used method to investigate solutions to the hybrid cloud challenge.

Advantages and disadvantages of cloud deployment models were summed up in a study [3] where the author alluded to the unique benefits as well as the complexity introduced by the combination of two or more unique cloud entities. He identified specific issues on network and data security owing to the increased attack surface area of the hybrid cloud and the ease with which sensitive data can be moved to the public cloud. The review did not mention intervention strategies however for the hybrid cloud challenges.

3 Key Method

This study employed a qualitative meta-analysis to aid in interpreting various themes emerging from hybrid cloud computing and its applications. The analysis followed a concept-centric approach to unpack literature and capture an interpretation that is representative of the whole field. The grounded theory facilitates the coding and interpretation of past studies [11] and employed in theming the body of research discovered. In the systematic review the following four broad methodologies were employed [12]: 1) data collection — database search; 2) selection —- application of inclusion and exclusion criteria; 3) coding —- categorization of attributes of the study; 4) synthesis — involving the analyses of results.

3.1 Data Collection

The databases employed in the data collection exercise together with the number of relevant studies selected are shown in Tab. 1. The search criteria was specified initially for the ACM Digital Library and later adapted for the other databases. The list of search keywords is shown in Appendix 6. In specifying the search keywords, consideration was given to studies that used hybrid cloud in some form of experiment and would be less successful otherwise. The results from the database search were pulled into the Mendeley[4] referencing tool and duplicates removed. The search keywords included “hybrid cloud” in

Table 1: Search results from selected databases

Database	Number of Results	Selected Results
ACM Digital Library	68	14
IEEE Xplore	144	20

the title of the study and any of the other keywords – economic, architecture, optimization, strategy, evaluation, adoption, selection, service, option, provisioning and model – had to appear in the abstract. In order to arrive at the set of search keywords, a preliminary search was done with “hybrid cloud” to give insight into the subject headings of the field of hybrid cloud computing. Some databases allowed for synonyms and wildcards to be used to further expand or refine the search. Where very few results were obtained, the search criterion was relaxed to bring out more results. It was discovered that the word “hybrid” referred to a mix of two or more systems or interventions.

An added method employed was the inspection of the references of selected studies for titles that contained “hybrid cloud”. This yield some further results of four studies that were added to Mendeley and resulting duplicates removed.

3.2 Selection

After removing duplicates, the abstracts were inspected for relevance to the central question of applying hybrid cloud in solving a cloud problem or optimizing the hybrid cloud with performance and economic motivations. The studies not fitting the criteria in the research question were eliminated. The full-text of the remainder were then reviewed for conformance to the inclusion and exclusion criteria. To minimize bias, a co-reviewer scanned through the abstracts of all the results obtained after removing duplicates from the search. The studies selected by both reviewers were included for full-text review.

The inclusion criteria were the use of hybrid cloud computing to overcome challenges faced at the computing industry and optimization of hybrid cloud deployments with respect to performance and cost. Exclusion criteria attempted to eliminate works that were not hybrid cloud specific. The inclusion and exclusion criteria are shown in Tab. 2.

3.3 Coding and Quality Evaluation

The reviewers adopted an open-ended coding scheme that was revised with each selected study. A matrix was developed with a summary of the studies under the headings of purpose, architecture or framework, category of hybrid

[4]http://www.mendeley.com

Table 2: Inclusion and Exclusion Criteria

Criteria Type	**Specification**
Inclusion	1. Studies that identify strategies for overcoming hybrid cloud specific challenges
	2. Experiments that aim at improving the performance, reduce cost or increase data security in the hybrid cloud
Exclusion	1. Works that capitalize on the benefits offered by the hybrid cloud to solve existing computing problems
	2. Works that are not specific to hybrid clouds but are generalized to cloud computing

cloud, method, experiment done and results achieved. Tab. 3 - Tab. 6 provide a summary of the findings from the synthesis.

4 Results

The two primary databases that were used were the ACM Digital Library and IEEE Xplore. A large number of the articles in the ACM Library were also discovered in IEEE Xplore. The number of studies that were found from the initial search was 212 and after removing duplicates and applying inclusion and exclusion criteria on the abstract and full-text, a total of 34 articles were selected and synthesized as follows:

- 12 articles focused on privacy and security of the sensitive data kept at the private side of the hybrid cloud
- 7 articles presented mathematical models for efficiently allocating resources (compute and network) under a time or cost constraint
- 15 articles were studies on optimizing the availability of resources via load balancing and task scheduling. Jobs are shifted around in public or private clouds depending on the demand for the particular type of workload.

Table 3: Summary of Studies Reviewed(1)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[13]	A MapReduce implementation to improve security during data processing in hybrid clouds.	privacy; security; performance	private, public cloud,Map Reduce	Experimental - involving 18 nodes. During the computation cycle implementation ensures not to leave sensitive data either in storage or detectable during execution.	Splitting data in to sensitive and non-sensitive sets and computing operations done at the public cloud with redundant computations in the private cloud	SEMROD achieved rates up to 4.6 times faster performance than other MR frameworks in the hybrid cloud
[16]	develop a method to handle data-intensive jobs with optimal cost load balancing for the hybrid cloud computing environment. Establish an equilibrium between cost and performance	cost optimization; performance; through load balancing	private, public cloud with Eucalyptus with three components - Cloud, Cluster and Node controllers	Experimental - involving 2 cloud systems. CPU usage and length of queue for disk processing determine load	middleware using method to perform load balancing by placing jobs in instances whilst watching for load approaching threshold values.	the experiment generated a pareto optimal curve indicating the threshold values to set for performance-cost load balancing in a hybrid cloud
					Eucalyptus as management and Dummynet to offer simulated ISP connection between public and private clouds	

Table 4: Summary of Studies Reviewed(2)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[18]	Presentation of a flexible and cost-effective service partitioning solution to the challenge of handling unpopular live-broadcasting channels in CLS systems	optimisation of broadcast latency	Amazon m3.large EC2 instances	Experimental -Place unpopular broadcasts in the public cloud and popular one on the datacentre to ensure low latency	1) Initial offloading based on the history of broadcaster, 2) ingesting redirection based on broadcasters performance 3) transcoding schedule available	HyCLS achieved cost savings of up to 19.5% and 20.4% when using only computations and views-based approach respectively
[14]	Proposal for a Smart Load Balancer and Bandwidth Shaper that keeps sensitive data in private cloud	security; privacy	public, off-premise private cloud; OpenStack	Proposed: controller, network and compute nodes	algorithm built inside the smart load balancer and bandwidth shaper routes data to appropriate network	Proposed architecture
[15]	Enhance security by partitioning an application automatically employing static code analysis and developer low-effort input	security; privacy; partitioning	private; public and multi-cloud; Java; PHP	Experimental:	free annotation of code units provide cues to an algorithm that employs dependency detection to cascade label other	algorithm worked for partitioning and deployment with low effort from developers
[19]	Develops a mathematical model for investment decisions on hybrid cloud computing	Pricing; Resource Allocation	public and private Mathematical Formula based on assumptions from M. Henneberger, 2016	Theoretical base on probability distribution of computing demands	find the minimum turning point on cost function	base model formulated but much will depend on the individual measurements taken

Table 5: Summary of Studies Reviewed(3)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[17]	Description of a resource allocation framework supporting a set of known time and cost constraints	economic; resource allocation	public and private Numerical Method	Experimental: employing the amazon auto scaling feature to increase or reduce nodes depending on constraint set	Run various jobs, systematic changing of input parameters until there is a convergence within the set constraint	the system was able to meet set constraints of cost and time with error margins of below 1.2% and 3.6% respectively
[20]	Description of a hybrid cloud storage system that tolerates public cloud outages with minimal overhead.	reliability, performance, storage cost and erasure coding	public and private;Key value store API from Apache Zookeeper. Metadata was stored as Zookeeper nodes. Openstack and Memcached were used in their default configuration	Experimental: Replicate data on f+1 clouds to mask faults on up to f clouds	Keep metadata in private cloud allowing more control, reducing latency and storage cost with guarantee for consistency	both microbenchmarks and YCSB macro benchmarks applied in the evaluation of the system showed reduced cost and higher consistency of storage
[21]	Introduce a privacy-aware computation framework based on MapReduce that is able to efficiently handle data-intensive workloads in a hybrid cloud	privacy; security; performance	public and private; MapReduce using Hadoop with prototype on FutureGrid	Experimental: employ a modified MapReduce framework	Planned placement of data and non-sensitive data processed in the public cloud	Privacy was preserved, scalability maintained and legacy computing jobs accommodated through the proposed framework

Table 6: Summary of Studies Reviewed(4)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[22]	to propose an approach to determining the impact of additional service to workflows in a hybrid cloud applying the maximum concurrent flow problem	performance, workflow deployment	public and private; Linear programming of the NAID algorithm	experimental: comparison of NAID algorithms for different networks and the amount of service workflows that can be provisioned	employing linear programming to determine the workflows that satisfy required percentage of the requested demand	solution of the NAID problem resulted in every workflow realizing an equal share of the global network demand. Adjusting individual shares also maximized the flow
[23]	presented an analysis of challenges experience during cloud bursting and suggested strategies and an experimental framework to deal with the challenges	cloud bursting; performance	private and public; MapReduce; OpenStack Icehouse	Proposed Experiment: In the Hadoop system set two racks for on-premise and off-premise to enable rebalancing to happen off-premise	leverage HDFS rebalancing mechanism to achieve high data locality for the MapReduce application. A policy of scheduling map tasks only on racks that have data migration completed	complementary strategies that are non-invasive and work out of the box
[26]	proposes an economic model that improves on costs	auction and markets	public and private;CloudSim	Experiment: Simulated public clouds and a million users with enhanced algorithms	The list of available VMs are provided via a broker to service users' requests based satisfaction of deadline and budget requirements	the enhanced economic model performed marginally better in cost validating goal of at least making it work

5 Analysis of Results

In analyzing the studies selected, an attempt was made to identify a main theme that runs through each one. Three main themes emerged even though more than one theme could be found in a study. The major themes emerging from the results centered on privacy and security (12 out of 34), improving efficiency of resource allocation (7 out of 34) and reducing the cost of processing workloads within a specified time constraint (15 out of 34). In the following sections, each theme is elaborated upon and subthemes introduced to further categorize the major theme.

5.1 Privacy and Security

Among the studies on privacy and security, twelve dwelt on data security and privacy. It can be noted that majority of the privacy and security studies viewed the hybrid cloud as the deployment model of choice to protect sensitive data and yet have the elasticity of the public cloud. Oktay et al. [13] split workload data into sensitive and non-sensitive sets, processing the sensitive part in the private cloud. Using a modified MapReduce framework, they achieved a processing rate 4.6 times faster than other MapReduce frameworks. Their goal was to ensure no trace of sensitive data was left in the public side of the hybrid cloud during processing. The elasticity of the public cloud was leveraged during the computation cycle. Cushman et al. [14] in their proposed Smart Load Balancer and Bandwidth Shaper (SLBBS) cloud employed this technique to keep sensitive data in the private cloud. The SLBBS detects and routes the data into an appropriate network based on sensitivity of the data. On the application front for hybrid clouds, [15] proposed to increase data security by partitioning the application to either run in the public or private side of the hybrid cloud. The application was split at the functional level using a dependency analysis of initially coded functions. In their experiments, it was tested and verified if privately originating data ever crossed over to the public side of the hybrid cloud. With functions calling sensitive data that reside on the private side, the data call also remained within the private boundaries of the network.

One recurring subtheme from the synthesis of hybrid cloud privacy and security studies was the need to split either the application or data and employing an algorithm to automate this process as much as possible. The performance of the algorithm was in most cases taken into consideration to ensure bottlenecks from overhead processing not degrade the overall efficiency of splitting the data or partitioning the application.

A second subtheme was the modification of an existing cloud computing framework in order to adapt it to the special case of the hybrid cloud. In [13], the MapReduce[5] framework was modified to process data that had been split

[5] https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

into two sets ensuring processing did not expose sensitive dataset in either storage or transit. Cushman et al. [14] employed the popular and open-sourced OpenStack[6] cloud computing framework and modified the network module with a built-in algorithm to detect and route data based on their sensitivity tagging. Smit et al. [15] facilitated the application partitioning via an algorithm embedded in a custom library of Annotations[7] and modified a Dependency Finder[8] to operate as a facilitating API. Their PHP application employed a tool called pCallGraph[9] for dependency analysis.

5.2 Cost Optimization

Four studies examined the cost element hybrid cloud computing through experiments in load balancing and resource allocation. Load balancing refers to routing workloads to facilitate a more even distribution among computing resources to enhance performance that will not be achieved otherwise. The elastic limitations of the private cloud infrastructure warrants heavy workloads are routed to public networks for handling at cost. This speeds up processing of the incoming workload but racks up an expense proportional to the workload process in the public cloud. To establish equilibrium between cost and performance, a fine line has to be pursued with the help of middleware that can maintain workloads within economic thresholds. Kasae & Oguchi [16] developed a middleware in their study to handle data-intensive jobs at optimal cost of load balancing and with the aim of achieving equilibrium between cost and performance. This was achieved by monitoring the CPU load levels and the queues for disk processing to determine if additional computing instances should be spawned on the public cloud.

The technique of handling workloads is a recurring theme in optimizing performance within cost constraints. For instance, [17] in their study set constraints of time and cost to their numerical method-based experiment. In it they run various workload sizes, systematically changing the input parameters of nodes and desired completion time until the arrival at an optimal set of parameters that fit a given constraint of either time or cost. Depending on the time or cost constraint, the number of nodes were adjusted accordingly. The very nature of the problem is solvable mathematically using models that have both predictive and optimization attributes as evidenced by [19] and [34].

Lee [19] developed a mathematical model for investment decision on hybrid cloud computing. The mathematical model had it's theoretical foundations in the probability distribution of computing demands. In this theme of optimizing cost or performance, the mathematical models offer minimum values which are then used to make decisions about the number of nodes to instantiate.

[6]https://www.openstack.org/
[7]https://docs.oracle.com/javase/tutorial/java/annotations/
[8]http://depfind.sourceforge.net
[9]http://phpcallgraph.sourceforge.net

Zhang, Liu & Wang [18] optimized the hybrid cloud for performance in a study about Crowdsourced Live Streaming where unpopular broadcasts were shifted programmatically to public infrastructure and highly subscribed broadcasts routed to the private cloud. The private cloud had a fixed cost and therefore could ran heavily subscribed broadcasts at little additional cost per new request. The element of history of the broadcaster served as an input to predict how heavily subscribed the channel will be in the future. The predictive element of the mathematical model feeds on the past history of the workload characteristics.

Ghouchani et al. [26] approached the issue of cost optimization through an economic model that leveraged auctions and markets in the public cloud service provision space. This allowed virtual machines to be allocated at the best (least) prices during cloud bursting.

5.3 Performance Enhancement

one of the main benefits of adopting a hybrid cloud is to improve the performance of the private infrastructure when workloads exceed the optimal threshold. The process of moving workloads into the public cloud often involves a communication cost, increased waiting time and security risks. This is especially true for data-intensive workloads, a scenario investigated in several studies [20]– [23] that try to find an optimal balance between performance cost and security [21] introduced a privacy-aware computation framework to efficiently handle data-intensive workloads in a hybrid cloud and [22] proposed an algorithm to minimize the impact of additional tasks on workflows.

Moving data within the hybrid cloud has a performance penalty, the severity of which depends on the processing location. Clemente-Castelló [23] and [20] explored a cloud storage system that tolerated public cloud outages, and leveraging the HDFS rebalancing mechanism to achieve high data locality respectively.

Task scheduling is an oft-studied area in cloud computing becomes more challenging in a hybrid cloud when performance is constrained by costs. Task scheduling studies attempt to find novel algorithms that reduce the total time used for processing assigned workloads whilst improving cost savings [32], [33]. A popular toolkit employed in cloud performance experiments is CloudSim, which is able to simulate public and private clouds with a static or variable channel between them. This often allows theoretical models to be almost operationalized for near-real-world testing. Fault tolerance is also a performance feature in a hybrid cloud. With the public cloud acting often as a backup, its availability and readiness impacts on hybrid cloud performance during demand spikes for computing resources [23], [28], [30]. Efficient algorithms pick out available spot instances in the public cloud to facilitate optimal job scheduling during cloud bursting. Not only do the algorithms help to efficiently schedule to use public clouds, they also dynamically scale virtual resources to meet processing deadlines of workload jobs [24], [35]. In scaling up virtual resources cost constraints are blended with deadlines to determine optimal allocation. Balagoni & Rao [27] experimented on scheduling using a simulation of 100 application that arrive with a Poisson distribution. They propose an algorithm that employs Weibull and normal distributions to efficiently schedule workloads with deadline and cost constraints.

The preceding sections presented the analysis of the results of the systematic literature review. The goal of the review was to find studies that addressed the challenges facing hybrid cloud adopters and categorize these studies into themes.

5.4 Limitations

Though the literature review was done with a systematic process of eliminating unconnected studies, there is a possibility of bias in the selection process as a result of the size of the team of researchers involved. This poses a threat to validity of the results which could not be overcome within the context of authors' environment of research. Much care has however been taken to eliminate bias in the database search and selection process. Two major databases were selected for the study and it is possible that some relevant studies might have been overlooked due to specification of a search criteria that will not produce them in initial results. Other databases were not searched due to human resource constraints and this is likely to impact on the conclusions drawn in this study.

6 Conclusion

Hybrid clouds, which are typically a combination of public and private clouds, offer an ideal environment for handling unpredictable demands for computing resources. The special challenges introduced as a result of combining two or more unique cloud deployment models have received significant but diverse attention in research resulting in an almost incoherent outlook for hybrid cloud computing. Still in its nascent stages, hybrid clouds have the potential of transforming enterprise IT, especially in small and medium enterprises. However, the seeming lack of coherent themes in hybrid cloud studies is a recipe for confusion for enterprise leaders who need clear strategies for adopting hybrid cloud services. This paper contributes to the synthesis of themes arising from characteristics of the hybrid cloud. The paper attempted to unpack the body of hybrid cloud computing literature to bring out thematic areas through a qualitative systematic review. The systematic review was carried out using a strict inclusion and exclusion criteria that helped to focus on studies related to strategies applied in hybrid cloud specific challenges. The study produced three thematic areas of interventions: data security, cost optimization and performance enhancement. A total of 34 studies were synthesized resulting in three thematic areas of data security, cost optimization and performance enhancement. The study showed that majority of research on hybrid cloud challenges belonged to one or more of these thematic areas.

References

[1] N. Ullah and S. U. Khan, “Challenges in the adoption of hybrid cloud: an exploratory study using systematic literature review,” J. Eng., 2016.

[2] David Roe, “Meet Forrester’s Hybrid Cloud Management Leaders,” 2016. [Online]. Available: https://www.cmswire.com/information-management/meet-forresters-hybrid-cloud-management-leaders/. [Accessed: 09-Feb-2018].

[3] S. Goyal, “Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review,” Int. J. Comput. Netw. Inf. Secur., vol. 6, no. 3, pp. 20–29, 2014.

[4] J. Weinman, “Intercloudonomics: Quantifying the Value of the Intercloud,” IEEE Cloud Comput., vol. 2, no. 5, pp. 40–47, Sep. 2015.

[5] J. Weinman, “Hybrid Cloud Economics,” IEEE Cloud Comput., vol. 3, no. 1, pp. 18–22, Jan. 2016.

[6] B. Martens, M. Walterbusch, and F. Teuteberg, “Costing of Cloud Computing Services: A Total Cost of Ownership Approach,” 45th Hawaii Int. Conf. Syst. Sci., pp. 1563–1572, 2012.

[7] D. S. Linthicum, “Emerging Hybrid Cloud Patterns,” IEEE Cloud Comput., vol. 3, no. 1, pp. 88–91, Jan. 2016.

[8] N. Chopra and S. Singh, “Survey on scheduling in hybrid clouds,” in Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1–6.

[9] F. Guenane et al., “Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture; Performance Model of MapReduce Iterative Applications for Hybrid Cloud Bursting,” in 2014 Global Information Infrastructure and Networking Symposium (GIIS); IEEE Transactions on Parallel and Distributed Systems, vol. PP, no. 99, p. 1; 1-6; 1.

[10] A. Bhardwaj, G. V. B. Subrahmanyam, V. Avasthi, and H. Sastry, “Three Tier Network Architecture to Mitigate DDoS Attacks on Hybrid Cloud Environments,” in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, 2016, p. 109:1–109:7.

[11] M. Weed, “Meta Interpretation: A Method for the Interpretive Synthesis of Qualitative Research,” Forum Qual. Soc. Res., vol. 6, no. 1, Jan. 2005.

[12] M. Borrego, M. J. Foster, and J. E. Froyd, "What Is the State of the Art of Systematic Review in Engineering Education?," J. Eng. Educ., vol. 104, no. 2, pp. 212–242, 2015.

[13] K. Y. Oktay, S. Mehrotra, V. Khadilkar, and M. Kantarcioglu, "SEMROD: Secure and Efficient MapReduce Over HybriD Clouds," in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, pp. 153–166.

[14] I. Cushman, L. Chen, D. B. Rawat, and N.-A. Le-Khac, "Designing Hybrid Cloud Computing Framework Using OpenStack for Supporting Multimedia with Security and Privacy," in Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, 2016, pp. 184–189.

[15] M. Smit, M. Shtern, B. Simmons, and M. Litoiu, "Partitioning Applications for Hybrid and Federated Clouds," in Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, 2012, pp. 27–41.

[16] Y. Kasae and M. Oguchi, "Proposal for an Optimal Job Allocation Method for Data-intensive Applications Based on Multiple Costs Balancing in a Hybrid Cloud Environment," in Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, 2013, p. 5:1–5:8.

[17] T. Bicer, D. Chiu, and G. Agrawal, "Time and Cost Sensitive Data-Intensive Computing on Hybrid Clouds," in Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012), 2012, pp. 636–643.

[18] C. Zhang, J. Liu, and H. Wang, "Towards Hybrid Cloud-assisted Crowdsourced Live Streaming: Measurement and Analysis," in Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2016, p. 1:1–1:6.

[19] I. Lee, "Determining an Optimal Mix of Hybrid Cloud Computing for Enterprises," in Companion Proceedings of the10th International Conference on Utility and Cloud Computing, 2017, pp. 53–58.

[20] P. Viotti, D. Dobre, and M. Vukolić, "Hybris: Robust Hybrid Cloud Storage," Proc. ACM Symp. Cloud Comput., vol. 13, no. 3, p. 12:1–12:14, Sep. 2014.

[21] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: Privacy-aware Data Intensive Computing on Hybrid Clouds," in Proceedings of the 18th ACM Conference on Computer and Communications Security, 2011, pp. 515–526.

[22] H. Moens, E. Truyen, S. Walraven, W. Joosen, B. Dhoedt, and F. De Turck, “Network-aware Impact Determination Algorithms for Service Workflow Deployment in Hybrid Clouds,” in Proceedings of the 8th International Conference on Network and Service Management, 2013, pp. 28–36.

[23] F. J. Clemente-Castelló, B. Nicolae, R. Mayo, J. C. Fernández, and M. M. Rafique, “On Exploiting Data Locality for Iterative Mapreduce Applications in Hybrid Clouds,” in Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, 2016, pp. 118–122.

[24] J.-W. Park, J. K. Yeom, J. Jo, and J. Hahm, “Elastic Resource Provisioning to Expand the Capacity of Cluster in Hybrid Computing Infrastructure,” in Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014, pp. 780–785.

[25] Y. Ahn and Y. Kim, “Auto-scaling of Virtual Resources for Scientific Workflows on Hybrid Clouds,” in Proceedings of the 5th ACM Workshop on Scientific Cloud Computing, 2014, pp. 47–52.

[26] B. E. Ghouchani, A. Abdullah, N. A. W. A. Hamid, and A. R. A. Rahiman, “A cost aware commodity market approach for real-time workload in Hybrid cloud,” in 2015 9th Malaysian Software Engineering Conference (MySEC), 2015, pp. 117–122.

[27] Y. Balagoni and R. R. Rao, “A cost-effective SLA-aware scheduling for hybrid cloud environment,” in 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, pp. 1–7.

[28] M. Lu and H. Yu, “A Fault Tolerant Strategy in Hybrid Cloud Based on QPN Performance Model,” in 2013 International Conference on Information Science and Applications (ICISA), 2013, pp. 1–7.

[29] G. Breiter and V. K. Naik, “A framework for controlling and managing hybrid cloud service integration,” in Proceedings of the IEEE International Conference on Cloud Engineering, IC2E 2013, 2013.

[30] R. Raju, J. Amudhavel, M. Pavithra, S. Anuja, and B. Abinaya, “A heuristic fault tolerant MapReduce framework for minimizing makespan in Hybrid Cloud Environment,” in 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 2014, pp. 1–4.

[31] U. Ganguly and C. Ray, “A secured and cost-effective method for processing queries using cloud resources optimally at hybrid cloud,” in 2016 International Conference on Computer, Electrical & Communication Engineering (ICCECE), 2016, pp. 1–8.

[32] W. Z. Jiang and Z. Q. Sheng, “A New Task Scheduling Algorithm in Hybrid Cloud Environment,” in 2012 International Conference on Cloud and Service Computing, 2012, pp. 45–49.

[33] H. Wei and F. Meng, “A Novel Scheduling Mechanism For Hybrid Cloud Systems,” in 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016, pp. 734–741.

[34] L. Zuo, L. Shu, S. Dong, Y. Chen, and L. Yan, “A Multi-Objective Hybrid Cloud Resource Scheduling Method Based on Deadline and Cost Constraints,” IEEE Access, vol. 5, pp. 22067–22080, 2017.

[35] Y. Ahn and Y. Kim, “VM Auto-Scaling for Workflows in Hybrid Cloud Computing,” in 2014 International Conference on Cloud and Autonomic Computing, 2014, pp. 237–240.

List of Search Keywords and wildcards employed

Keywords option evaluation service provisioning model architecture adoption selection strategy strategies economic optimization

Wildcards adopt*, architect*, performance, optim*, economic*, cost*, strateg*

Sample search criteria in IEEE Xplore ((“Document Title”:hybrid cloud) AND ((“Abstract”:economics) OR (“Abstract”:architecture) OR (“Abstract”:strategy) OR (“Abstract”:evaluation) OR (“Abstract”:optimization) OR (“Abstract”:adoption)))

SUPPLEMENTARY MATERIAL - PAPER B

Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis

Summary of Studies Reviewed

References have been listed at the end of this supplementary section to support the reading of the Tables SS1 - SS4.

Table SS1: Summary of Studies Reviewed(a)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[24]	present an architecture with dynamic allocation of virtual machines based on demand threshold.	cloud bursting; performance	private and public; physical and virtual nodes together in HIVE with VLAN configuration. Message-Passing Interface (MPI) service or High Throughput Computing (HTC) service. Tools include OpenNebula, OpenMPI, High-Performance Linpack and HTCondor	Experiment: On-Demand Cluster Provider categorises the scientific workflow into MPI or HTC; Dynamic Infrastructure Provider - allocate and monitor computing resources; and Service-aware Job Manager - determined appropriate job scheduler for the hybrid cluster	Monitoring of datacentre sites for queues that exceed a given level within a given time and the automatic creation of VMs if values over the threshold. VMs in cloud are connected to physical nodes by configuring VLAN	though VMs take time to generate but once ready, upto 64 VMs were managed in less than 4 minutes

Table SS2: Summary of Studies Reviewed(b)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[25]	presents an auto-scaling method with priority consideration for deadlines by maximising the allocation of resources	resource allocation; auto-scaling	public and private;CloudSim with four private clouds and Amazon EC2	Experiment employing an auto-scaling algorithm on Protein annotation workflow focusing on the input and output data and their transfer time	automatic scale-in and scale-out based on task dependency and data transfer time in order to complete workflow within deadline. Private cloud VMs are spawned until bursting applied to accommodate time constraint	all tasks were finished within deadline at maximum efficiency and within the time constrained priority
[27]	Proposes an algorithm to efficiently schedule workload with deadline and cost constraints	performance and cost optimization	private and public; CloudSim with Amazon EC2 spot instances	Experiment: heterogeneous nodes and synthetic workloads	simulated 100 applications arrive following Poisson distribution. Weibull and normal distribution used for the workload distribution	the scheduling algorithm completed the tasks within deadline and at optimized cost
[28]	propose a fault tolerant strategy which takes into account performance and request completion of workloads	performance and fault tolerance	public and private;Queuing Petri net Modeling Environment (QPME), SimQPN	Experiment using queuing Petri nets to model the hybrid cloud.	fault types analyzed and employed in determining performance of the platform	the strategy worked under both low and high load states of the hybrid cloud platform

Table SS3: Summary of Studies Reviewed(c)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[29]	present an architectural framework for hybrid cloud management and control based on policies	integration management and control	public and private on- and off-premise, and multicloud; IBM Service Management Extensions for Hybrid Cloud	Design of architecture to cater for identified integration challenges	Introduction of a hybrid cloud management layer having plugins for various services such as SaaS Apps and Monitoring	three types of service integration patterns - horizontal (homogeneous), vertical (heterogeneous) and management integration
[30]	present a strategy for speculative execution improvement by constructing an optimal job schedule for workloads	performance and time optimization	public and private; MapReduce using Hadoop framework on Ubuntu OS	Experiment: virtual machines with jobs arranged in random sequence and scheduled using Johnson Scheduling	Two steps in Optimal Time Algorithm 1) node prioritization 2) selection of the earliest time process	minimization of makespan - time to complete entire workload - of DAG MapReduce algorithm is realized
[31]	present a model for maintaining privacy of sensitive data at an optimized cost within a hybrid cloud employing an authentication monitor	data privacy with cost optimization	public and private; mathematical model	Theoretically generated model with a set of sample data input for testing and mainly measuring the amount of delay time for processing	delay-based cost model that decided where in hybrid cloud data processing is to take place	the location of data processing depended on the proportion of sensitive data in the workload
[32]	propose a graph-based task scheduling algorithm that minimizes cost of processing workloads in a hybrid cloud	task scheduling and cost optimization	public and private; CloudSim Toolkit; CloudReports	Experiment: simulation to employ graph-based task scheduling algorithm using a private cloud and a public cloud	Hopcroft-Karp algorithm employed in finds maximal set of shortest augmenting paths in bipartite graph	average improvement on cost savings compared with other algorithms on task scheduling

Table SS4: Summary of Studies Reviewed(d)

Study	Purpose	Category	Architecture/ Framework	Method	Experiment/ Intervention Description	Results / Findings
[33]	present a cloud management system with an innovative user request scheduling mechanism	management and task scheduling	public and private; one cloud provider, four clusters, each with four machine groups	Experiment: set of algorithms for single job scheduling, intra-cluster load and inter-cluster load balancing	introduction of load balancing at two tiers	makespan on average has 20% and 35 % reduction when intra-cluster load and inter-cluster balancing respectively is deployed

References

[1] N. Ullah and S. U. Khan, "Challenges in the adoption of hybrid cloud: an exploratory study using systematic literature review," J. Eng., 2016.

[2] David Roe, "Meet Forrester's Hybrid Cloud Management Leaders," 2016. [Online]. Available: https://www.cmswire.com/information-management/meet-forresters-hybrid-cloud-management-leaders/. [Accessed: 09-Feb-2018].

[3] S. Goyal, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review," Int. J. Comput. Netw. Inf. Secur., vol. 6, no. 3, pp. 20–29, 2014.

[4] J. Weinman, "Intercloudonomics: Quantifying the Value of the Intercloud," IEEE Cloud Comput., vol. 2, no. 5, pp. 40–47, Sep. 2015.

[5] J. Weinman, "Hybrid Cloud Economics," IEEE Cloud Comput., vol. 3, no. 1, pp. 18–22, Jan. 2016.

[6] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of Cloud Computing Services: A Total Cost of Ownership Approach," 45th Hawaii Int. Conf. Syst. Sci., pp. 1563–1572, 2012.

[7] D. S. Linthicum, "Emerging Hybrid Cloud Patterns," IEEE Cloud Comput., vol. 3, no. 1, pp. 88–91, Jan. 2016.

[8] N. Chopra and S. Singh, "Survey on scheduling in hybrid clouds," in Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1–6.

[9] F. Guenane et al., "Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture; Performance Model of MapReduce Iterative Applications for Hybrid Cloud Bursting," in 2014 Global Information Infrastructure and Networking Symposium (GIIS); IEEE Transactions on Parallel and Distributed Systems, vol. PP, no. 99, p. 1; 1-6; 1.

[10] A. Bhardwaj, G. V. B. Subrahmanyam, V. Avasthi, and H. Sastry, "Three Tier Network Architecture to Mitigate DDoS Attacks on Hybrid Cloud Environments," in Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, 2016, p. 109:1–109:7.

[11] M. Weed, "Meta Interpretation: A Method for the Interpretive Synthesis of Qualitative Research," Forum Qual. Soc. Res., vol. 6, no. 1, Jan. 2005.

[12] M. Borrego, M. J. Foster, and J. E. Froyd, "What Is the State of the Art of Systematic Review in Engineering Education?," J. Eng. Educ., vol. 104, no. 2, pp. 212–242, 2015.

[13] K. Y. Oktay, S. Mehrotra, V. Khadilkar, and M. Kantarcioglu, "SEMROD: Secure and Efficient MapReduce Over HybriD Clouds," in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015, pp. 153–166.

[14] I. Cushman, L. Chen, D. B. Rawat, and N.-A. Le-Khac, "Designing Hybrid Cloud Computing Framework Using OpenStack for Supporting Multimedia with Security and Privacy," in Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, 2016, pp. 184–189.

[15] M. Smit, M. Shtern, B. Simmons, and M. Litoiu, "Partitioning Applications for Hybrid and Federated Clouds," in Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, 2012, pp. 27–41.

[16] Y. Kasae and M. Oguchi, "Proposal for an Optimal Job Allocation Method for Data-intensive Applications Based on Multiple Costs Balancing in a Hybrid Cloud Environment," in Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, 2013, p. 5:1–5:8.

[17] T. Bicer, D. Chiu, and G. Agrawal, "Time and Cost Sensitive Data-Intensive Computing on Hybrid Clouds," in Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012), 2012, pp. 636–643.

[18] C. Zhang, J. Liu, and H. Wang, "Towards Hybrid Cloud-assisted Crowdsourced Live Streaming: Measurement and Analysis," in Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2016, p. 1:1–1:6.

[19] I. Lee, "Determining an Optimal Mix of Hybrid Cloud Computing for Enterprises," in Companion Proceedings of the10th International Conference on Utility and Cloud Computing, 2017, pp. 53–58.

[20] P. Viotti, D. Dobre, and M. Vukolić, "Hybris: Robust Hybrid Cloud Storage," Proc. ACM Symp. Cloud Comput., vol. 13, no. 3, p. 12:1–12:14, Sep. 2014.

[21] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: Privacy-aware Data Intensive Computing on Hybrid Clouds," in Proceedings of the 18th ACM Conference on Computer and Communications Security, 2011, pp. 515–526.

[22] H. Moens, E. Truyen, S. Walraven, W. Joosen, B. Dhoedt, and F. De Turck, "Network-aware Impact Determination Algorithms for Service Workflow Deployment in Hybrid Clouds," in Proceedings of the 8th International Conference on Network and Service Management, 2013, pp. 28–36.

[23] F. J. Clemente-Castelló, B. Nicolae, R. Mayo, J. C. Fernández, and M. M. Rafique, "On Exploiting Data Locality for Iterative Mapreduce Applications in Hybrid Clouds," in Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, 2016, pp. 118–122.

[24] J.-W. Park, J. K. Yeom, J. Jo, and J. Hahm, "Elastic Resource Provisioning to Expand the Capacity of Cluster in Hybrid Computing Infrastructure," in Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014, pp. 780–785.

[25] Y. Ahn and Y. Kim, "Auto-scaling of Virtual Resources for Scientific Workflows on Hybrid Clouds," in Proceedings of the 5th ACM Workshop on Scientific Cloud Computing, 2014, pp. 47–52.

[26] B. E. Ghouchani, A. Abdullah, N. A. W. A. Hamid, and A. R. A. Rahiman, "A cost aware commodity market approach for real-time workload in Hybrid cloud," in 2015 9th Malaysian Software Engineering Conference (MySEC), 2015, pp. 117–122.

[27] Y. Balagoni and R. R. Rao, "A cost-effective SLA-aware scheduling for hybrid cloud environment," in 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, pp. 1–7.

[28] M. Lu and H. Yu, "A Fault Tolerant Strategy in Hybrid Cloud Based on QPN Performance Model," in 2013 International Conference on Information Science and Applications (ICISA), 2013, pp. 1–7.

[29] G. Breiter and V. K. Naik, "A framework for controlling and managing hybrid cloud service integration," in Proceedings of the IEEE International Conference on Cloud Engineering, IC2E 2013, 2013.

[30] R. Raju, J. Amudhavel, M. Pavithra, S. Anuja, and B. Abinaya, "A heuristic fault tolerant MapReduce framework for minimizing makespan in Hybrid Cloud Environment," in 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 2014, pp. 1–4.

[31] U. Ganguly and C. Ray, "A secured and cost-effective method for processing queries using cloud resources optimally at hybrid cloud," in 2016 International Conference on Computer, Electrical & Communication Engineering (ICCECE), 2016, pp. 1–8.

[32] W. Z. Jiang and Z. Q. Sheng, "A New Task Scheduling Algorithm in Hybrid Cloud Environment," in 2012 International Conference on Cloud and Service Computing, 2012, pp. 45–49.

[33] H. Wei and F. Meng, "A Novel Scheduling Mechanism For Hybrid Cloud Systems," in 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016, pp. 734–741.

[34] L. Zuo, L. Shu, S. Dong, Y. Chen, and L. Yan, "A Multi-Objective Hybrid Cloud Resource Scheduling Method Based on Deadline and Cost Constraints," IEEE Access, vol. 5, pp. 22067–22080, 2017.

[35] Y. Ahn and Y. Kim, "VM Auto-Scaling for Workflows in Hybrid Cloud Computing," in 2014 International Conference on Cloud and Autonomic Computing, 2014, pp. 237–240.

Paper C

Load balancing in hybrid clouds through process mining monitoring

Azumah, Kenneth Kwame, Kosta, Sokol and Sørensen, Lene Tolstrup

The paper has been published in the
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Vol. 11874 LNCS, pp. 148–157, 2019.

Abstract

An increasing number of organisations are harnessing the benefits of hybrid cloud adoption to support their business goals and achieving privacy and control in a private cloud whilst enjoying the on-demand scalability of the public cloud. However the complexity introduced by the combination of the public and private clouds worsens visibility in cloud monitoring with regards compliance to given business constraints. Load balancing as a technique for evenly distributing workloads can be leveraged together with process mining to help ease the monitoring challenge. In this paper we propose a load balancing approach to distribute workloads in order to minimise violations to specified business constraints. The scenario of a hospital consultation process is employed as a use case in monitoring and controlling Octavia load balancing-as-a-service in OpenStack. The results show a co-occurrence of constraint violations and Octavia L7 Policy creation, indicating a successful application of process mining monitoring in load balancing.

Keywords:

Hybrid cloud, process mining, event calculus, OpenStack Octavia

1 Introduction

Lots of data generated in organisations help make decisions about their business. In the healthcare industry the data generally result from processes involving the patient and clinicians, encompassing a wide range of activities such consultations, appointments booking, laboratory and radiological investigations. The collection, processing, and storage of such data present a notable challenge including the extent of IT investment needed, in the form of infrastructure, to ensure that adequate mileage is obtained from the data being generated. Though Cloud Computing eases the IT infrastructure decision-making and risks, the concerns about privacy and security continue to impact on the extent of adoption. Businesses generally protect their sensitive information by storing and processing data in private data centres, where full control offers a high level of assurance of privacy and preservation of business information. It is therefore desirable for companies to keep sensitive data on an internal data centre and process less sensitive, irregular or seasonal workloads in the public cloud. The combination of the public cloud and private data centre is often termed a *hybrid cloud*, with the public cloud providing high scalability potential whilst the private data centre provides privacy for selected data and applications. As a result of the combination of two separate deployment models, a hybrid cloud increases the complexity of managing the overall cloud infrastructure, especially in maintaining privacy and control over data and processes.

Load balancing is one of the major cloud computing strategies [1], designed to help address the challenge of maintaining adequate visibility and control over workload distribution within the cloud. Load balancing is applied in situations requiring high availability of a service — especially in a hybrid cloud where resources are split between one or more private data centres and the public cloud. Basic routing of network packets enables messages to be distributed at the media level (OSI Layer 3) however for more data-aware routing, the host level (OSI Layer 7) is employed in network packet distribution. As a strategy for managing workloads in a hybrid cloud, load balancing employs various algorithms influenced by the industry's business processes or constraints. A business constraint commonly refers to anything that interferes with the profitability of a business endeavour. Among these constraints are company policies on data processing to increase some level of compliance to regulations in the industry. The increasing complexity and dynamic nature of business constraints, however, requires matching sophisticated and more intelligent algorithms to satisfy load balancing objectives. Programming such constraints to provide control and compliance to predefined rules will inherently be procedural. It will be unreasonable to require static verification techniques to define explicitly all possible allowable procedures. This complexity makes regular programming languages virtually unsuitable for monitoring event and process violations as they occur.

Process mining is a rising data-aware technique that has long been used to map out the process model of business activities and provide insight into compliance of the activities to some given process model. The technique employs event data obtained from the log files of the information system generating the activities. Existing solutions however present crucial limitations in a load balancing situation that requires near immediate reactions to service requests [2], [3]. This motivates a study linking process mining and load balancing to achieve near-realtime compliance to given constraints. In this paper, we present a prototype framework that employs layer 7 load balancing as a means of controlling data processing in a hybrid cloud to satisfy as much as possible a given set of business constraints. Our framework extends Octavia, an LBaaS module in OpenStack with a modified implementation of MoBuCon [4], a process mining tool. Our framework is able to dynamically update the load balancer policies based on event data and specified business constraints. The results show how a load balancer can route requests to avoid violating constraints that are dependent on the events occurring within a hybrid cloud.

The remaining sections are organised as follows: We first present background and related work on load balancing and monitoring business constraints via process mining (Sect.2). Sect.3 lays out the architecture and integration of the process mining and load balancing components, and also describes the environment of the experiment. Sect. 4 presents the preliminary results and Sect. 5 concludes the work.

2 Background and Related Work

A business constraint commonly refers to anything that interferes with the profitability of a business endeavour. This section provides a background on business constraints and how event data can influence load balancing behaviour.

2.1 Business Constraints and Event Calculus

Business constraints are characterised by rules that force an organisation's compliance to certain governing practices. For example, the handling of sensitive data should follow a given sequence within organisations' information systems. The analyses of log files containing event data therefore becomes a crucial aspect of monitoring information systems. Process mining as a data analysis technique relies on three key pieces of event data – **instance identifier**, **activity name** and **timestamp**. The timestamp provides a chronological order for the capture of activities occurring in information systems. Kowalski formalises the notion of an event in time employing the Horn clause with negation as failure [5]. The resulting event calculus formalisation is executable in a logical manner and therefore presentable as a Prolog program.

Event logs captured from information systems provide a record of activities that occurred in a chronological manner. A group of activities usually culminates in a business process designed to achieve one or more specific goals. By analysing the event logs for the correct order of occurrence of the activities and their duration, deductions can be made about whether the process complies with specified rules. Event Calculus (EC) is characterised by reasoning about change and the effects of an action on conditions that can change over time. EC is thus suitable for monitoring and reporting violations of policies by events occurring in the cloud infrastructure.

Business constraints – hereafter referred to as *constraints* – are expressed graphically using the Declare notation [6]. A constraint specified as ***response (A, B, Cond)*** is a Declare constraint with a data condition *Cond* that states "if event A occurs and Cond holds, event B must occur afterwards". The constraint is said to be *activated* when an event matching a part of the constraint is encountered in the log [7]. Given an event log with traces {A, B, C}, {A, B} and {A, C}, the last trace violates the constraint since B must necessarily follow A for compliance. The traces have the following states: $\{...A\} \rightarrow pend, \{...A, B\} \rightarrow sat, \{...A, B, C\} \rightarrow sat, \{...A, B, C, A\} \rightarrow pend, \{...A, B, C, A, C\} \rightarrow viol$. For each partial trace, the truth-value of the Linear Temporal Logic (LTL) formula [4] generates the state of satisfaction of the constraint, either satisfied *sat*, violated *viol* or pending *pend* [8]. Extending Declare with data-awareness facilitates the monitoring of business constraints [6], [9], [4].

2.2 Octavia - Load-balancing-as-a-service Overview

OpenStack started in 2010 as a joint project between NASA and Rackspace. Octavia, one of its key projects, provides scalable load-balancing-as-a-service (LBaaS) with layer 7 (L7) content switching capability at layer 2 speeds [10]. When users access services hosted on the cloud, their requests are usually handled by a load balancer. The load balancing algorithm is based on one or more statistics including CPU load, number of connections and affinity settings. Layer 7 Load Balancing is achieved through L7 rules and L7 policies. The **L7 Rule** is a simple, single logical test, which evaluates to either `true` or `false`. An **L7 Policy** is a collection of L7 rules and a defined action when all rules in the policy are matched.

Sharma et al. present a case for evenly spreading workloads across a set of available VMs [11]. In finding an optimal solution, they introduce an algorithm that mimics the way bats locate and identify their prey: bats representing workloads and targeted prey being the virtual machines. Wider issues depending on the content of the workload and data-awareness are not covered.

Rahhali [12] considers and efficient combination of algorithms to optimise energy consumption and response time – both essential QoS parameters in cloud computing. The paper employs well-known heuristics to achieve near-optimal solutions targeting both running time and energy consumption. Their algorithms do not cover data-awareness in the workloads.

Aktas [13] proposes a monitoring software architecture for both preventive maintenance and error detection in a cloud computing environment. The software uses primitive metric data gleaned from logs of the cloud computing operating system and utilises metrics on CPU, memory, and disk usage to trigger reports on measurements that violate preset rules. The paper shows the capacity of the software to handle large volumes of events but the monitoring rules are preset.

Liu and Li [14] present a stratified monitoring model for hybrid clouds and propose key metrics for each monitoring layer putting forth measures to perform as part of data collection for evaluation. The paper employs agent technology towards a monitoring architecture. There are no experiments on implementation and evaluation of the model and its efficiency.

Azumah et al. [15] have proposed a hybrid cloud scheduling with process mining monitoring mechanism that facilitates decision-making of the scheduler. They present their experiments with CloudSim [16], showing an output that employs Event Calculus in determining compliance or otherwise to set a set of given business constraints. The paper shows how process mining monitoring can influence scheduling in the hybrid cloud towards achieving a more desirable and proportionate VM spawning. Their experiment employ synthetic data in a simulated environment.

3 System Specification

3.1 Use Case Scenario

In this work, we employ a hospital scenario where L7 content switching can be of benefit when performing load balancing in a hybrid cloud. Recall one of the benefits of the hybrid cloud is the on-demand availability of public resources when the private datacenter has reached peak capacity. In our hospital scenario, the constraint for processing data in the hybrid cloud is re-routing, as much as possible, highly sensitive data *HS* to the internal datacenter *IDC*. The processing of less sensitive data *LS* can take place in either IDC or public portion of the hybrid cloud *EDC*. We consider the constraint *conHSCX* defined as follows: 1) If a patient is provisionally diagnosed with cancer, 2) Laboratory investigations reveal positive markers for cancer 3) The next consultation should be regarded as sensitive data processing and routed accordingly to the internal datacenter.

Table 1: Partial trace involving case IDs during cloud bursting.

#	Case ID	Activity	Timestamp	Route	Activation
L1	PatientA	Diagnose: Cancer	Jan 3, 2019 9:25	EDC	
L2	PatientB	Diagnose: Hypertension	Jan 3, 2019 10:05	EDC	
L3	PatientA	Lab: CA+	Jan 3, 2019 11:01	EDC	
L4	PatientC	Diagnose: Anaemia	Jan 3, 2019 11:02	EDC	
L5	PatientB	Lab: PSA+	Jan 3, 2019 11:30	EDC	
L6	PatientB	Diagnose: Cancer	Jan 4, 2019 10:08	EDC	
L7	PatientA	Consult: GP	Jan 9, 2019 9:02	IDC	conHSCX
L8	PatientA	Lab: FBS	Jan 9, 2019 10:42	IDC	
L9	PatientC	Lab: RBS	Jan 9, 2019 10:50	EDC	
L10	PatientB	Consult: Cancer	Jan 9, 2019 11:21	EDC	
L11	...	...	...	...	...

For instance, the partial trace in Table 1 shows the activation of the conHSCX constraint on account of PatientA having been provisionally diagnosed and then having lab results confirming the diagnosis. The confirmation of the lab results activates conHSCX and classifies subsequent events involving PatientA as highly sensitive as far as the hospital policy is concerned. In the next sections we treat the architectural layout for the evaluation and routing of application server requests based on Octavia L7 policies and rules.

3.2 Architectural Layout and Component Integration

We present the individual components of the architecture together with configuration and extensions that facilitate their integration.

In the architecture we parse event data generated by the load balancer, *HAProxy* and a simulated application server. To scale the architecture to contain more than one load balancer and application server, a log service *Logstash*

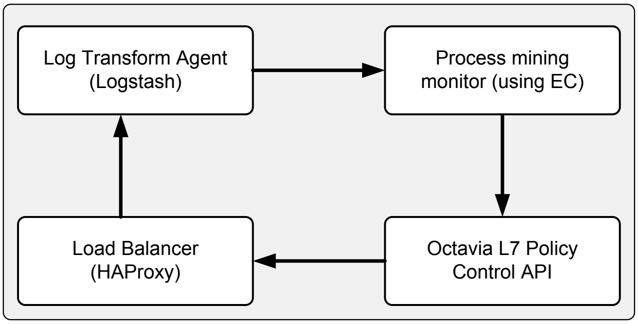


Fig. 1: Highlevel Architectural Representation with Process Flow

```
May 31 18:48:14 10.0.0.1:37318 0/0/4/3/16 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientA"
May 31 18:48:15 10.0.0.1:37320 0/0/3/2/16 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientB"
May 31 18:48:16 10.0.0.1:37322 0/0/14/18/48 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1 caseid:patientA"
May 31 18:48:18 10.0.0.1:37324 0/0/3/2/8 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientC"
May 31 18:48:19 10.0.0.1:37326 0/0/3/4/18 200 40 - - ---- 0/0/0/0/0 0/0 "GET / HTTP/1.1 caseid:patientA"
May 31 18:48:20 10.0.0.1:37328 0/0/3/2/18 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1 caseid:patientB"
```

Fig. 2: HAProxy Logging

is introduced to help aggregate event logs from the various VMs. We filter and transform the Logstash data into our *Process Mining Monitor* (PMM) for further evaluation. Further information is provided by configuring HAProxy to display custom headers using the built-in `http-format`. The application server within the hybrid cloud sends requests with custom headers containing information about activity, timestamp and case ID. By injecting custom headers into each request we enable the load balancer to make more "informed" decisions about routing dynamically. The flow of information between the components is shown in Fig. 1.

We have configured HAProxy to record in its log file each request on the HTTP or application layer that includes the URL requested and the server to which the request was routed alongside the custom headers. Before feeding the information as a message to our implementation of PMM, we extract and analyse the transformed fields: event type, activity name, case id, timestamp, source address and backend server. The source address and the backend machine servicing the request constitute extra resources that we use in the process mining stage. Taking one line in the HAProxy log – `May 31 18:48:14 10.0.0.1:37318 0/0/4/3/16 200 40 - - - - - - 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientA"` – we extract `patientA, /consult, May 31 18:48:14, 10.0.0.1` for input into PMM.

Fig. 2 shows the custom HAProxy log format that embeds header information on case ID, activity and timestamp. Similarly, the transformed HAProxy log shows event data adapted to the input of PMM – Fig. 3.

```
patientA, /consult, May 31 18:48:14, 10.0.0.1
patientB, /consult, May 31 18:48:15, 10.0.0.1
patientA, /lab,     May 31 18:48:16, 10.0.0.1
patientC, /consult, May 31 18:48:18, 10.0.0.1
patientA, /,        May 31 18:48:19, 10.0.0.1
patientB, /lab,     May 31 18:48:20, 10.0.0.1
```

Fig. 3: Extracted and Transformed HAProxy Log

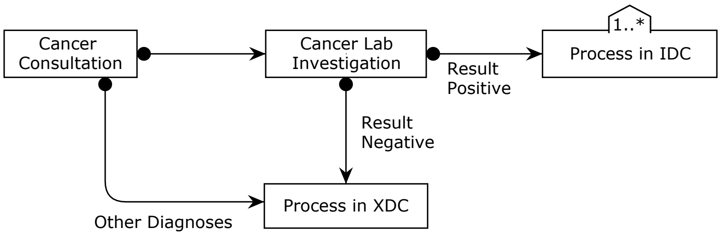


Fig. 4: Constraint Specification in Declare

3.3 Process Mining Monitor

PMM is a Python extension of the MoBuCon framework [4] that features a Prolog engine for logical evaluation of event traces. Our implementation has two interfaces: `constraint` interface for specification of Declare [17] constraints and `eventdata` interface for feeding events into the Prolog engine; and an output log showing violations that have occurred. We have specified the constraint using an external Declare system[1] and exported the configuration file onto the Octavia host machine to used by the PMM which also reads the event data as it occurs. The constraint specification shown in Fig. 4 represents the use case scenario described in Sect. 3.1 but relates to one case ID. It specifies that all activities after `Results Positive` processing should be confined to the internal datacenter. This means all other activities before such a diagnosis confirmation can be processed in any portion of the hybrid cloud *XDC*. A violation therefore occurs when a trace involving case ID has been processed in EDC after `Result Positive`. We programme PMM to make API calls to the Octavia Worker to create a new policy involving the violated case ID. The next section describes the call to the Octavia Worker API.

3.4 Octavia L7 Policy Control API

We considered all post-*Results Positive* events related to a case ID as highly sensitive (HS) and all others, less sensitive (LS). For HS events all requests are to be routed to the *IDC pool* of backend servers whilst all other traffic is routed to any available pool. We focused on creating L7 policies and rules only for the circumstance where HS traffic is routed to EDC. After encountering a violation

[1] https://www.win.tue.nl/declare/

involving a case ID, PatientA, say, we create a matching L7 policy to redirect further requests to IDC pool as follows: `openstack loadbalancer l7policy create -action REDIRECT_TO_POOL -redirect-pool IDC_pool -name policy01_PatientA listener_hybrid_cloud`. We then add the matching L7 rule to check whether header information contains the involved case ID as follows: `openstack loadbalancer l7rule create -compare-type CONTAINS -type HEADER -key caseid -value patientA policy01_PatientA`. With continued violations occurring for a case ID, `policy02_PatientA` will be the next L7 policy to be created having the same L7 rule as in `policy01_PatientA`. The combination of the two policies has no extra effect on load balancer since their L7 rules are the same. In our setup the listener `listener_hybrid_cloud` on the load balancer is the central port to which all application server requests are sent. It effectively distributes traffic using the ROUND_ROBIN algorithm by default and applies any attached L7 policies.

3.5 Setup and Experimentation

The experiment was ran on a four-core virtual machine with 16GB memory having Ubuntu 16.04 LTS as operating system (OS). We installed OpenStack using a set of extensible scripts from DevStack[2] (version *train*) to create the hypervisor on top of the Ubuntu OS. The load balancing-as-a-service module, Octavia, was enabled alongside the compute, networking and virtual machine imaging modules. Openstack served as our *hybrid cloud operating system* through which the VMs and load balancer were spawned. Three VMs running the *cirros m1.tiny* flavour of the linux OS each had 512MB RAM, 1GB allocated disk space and one VCPU. The load balancer VM running HAProxy had 1GB RAM, 2GB disk space and one VCPU. In a live environment, the requests from users hit the load balancer VM and a corresponding event log is kept by the information system showing a record of activities initiated. Because L7 policy impacts the behaviour of the load balancer, API calls to the Octavia Worker for policy changes must be influenced by the process mining monitor.

4 Results and Evaluation

The constraint violations detected by the PMM component trigger L7 policy changes to the load balancer. In our experiment, we measure the correlation of violations and the L7 policies created in the load balancer. This implies comparing the timestamps of occurrence of a violation and L7 policy creation gives an indication of the responsiveness of the overall system. Our experiment recorded 27 occurrences of violations from a synthetic event log. For each of the constraint violations, we detected a corresponding creation of an L7 policy and associated L7 rule. The average time for the creation of the policy and rule in our environment is 12.27 secs with the quickest time being 7.26 secs.

[2]https://docs.openstack.org/devstack/latest/

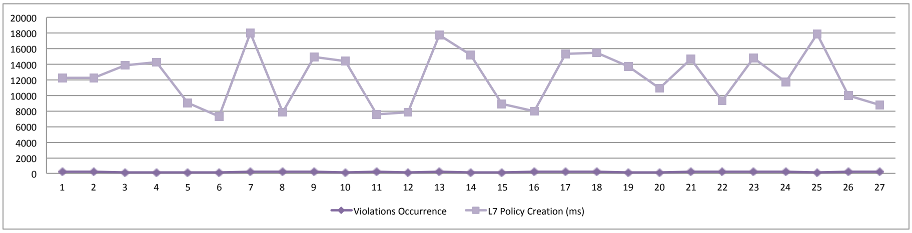


Fig. 5: Co-occurrence of Constraint Violations and L7 Policy Creation

The longest time taken to create a L7 policy and rule is measured at 17.95 secs. Figure 5 displays the co-occurrence of the constraint violations with the creation of L7 policies. There is a one-to-one correlation between the two factors validating a viable trigger through violation detection from process mining.

5 Conclusion and Future Work

Decisions for adopting hybrid Cloud Computing generally consider cost, security and privacy as primary factors. The availability of monitoring tools therefore facilitates the decision-making of an organisation towards workload distribution in a hybrid cloud to achieve a high compliance to business constraints. The distribution of the workload is often done via scheduling and load balancing mechanisms. In this paper we have looked at the load balancing-as-a-service mechanism of OpenStack's Octavia project and how it can be controlled through the process mining of event logs generated from the cloud information system. Taking advantage of the filtering mechanism built into Octavia, we triggered the control of the load balancer to re-route application server requests in line with given constraints. We expressed the constraints using the Declare language which enable an event trace to be evaluation either as true (compliance) or false (violation) by the Prolog-embedded MoBuCon process mining tool. Our experiment showed a one-to-one correspondence between violations and the creation of L7 policies of the load balancer to distribute application requests to the appropriate servers. This ultimately increases the compliance of data processing to the given business constraint.

Our setup allowed for one constraint to be programmed into the Octavia project. In the real world however, a business process typically involves multiple constraints to ensure compliance to some given objectives. To apply our setup to such situations therefore invites the control of multiple constraints with the possibility increasing time lags in overall load balancer response. This situation is worth exploring in the future where this work will be expanded to allow the specification of multiple constraints via the Declare [17] tool.

References

[1] K. K. Azumah, L. T. Sorensen, and R. Tadayoni, "Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis", in 2018 IEEE 7th International Conference on Adaptive Science and Technology (ICAST), 2018, pp. 1–8.

[2] R. Gandhi, Y. C. Hu, and M. Zhang, "Yoda: a highly available layer-7 load balancer", in Proceedings of the Eleventh European Conference on Computer Systems - EuroSys '16, 2016, pp. 1–16.

[3] N. Rathore, "Performance of Hybrid Load Balancing Algorithm in Distributed Web Server System", Wirel. Pers. Commun., vol. 101, no. 3, pp. 1233–1246, Aug. 2018.

[4] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. van der Aalst, "Monitoring business constraints with the event calculus", ACM Trans. Intell. Syst. Technol., vol. 5, no. 1, pp. 1–30, Dec. 2013.

[5] R. Kowalski and M. Sergot, "A logic-based calculus of events", New Gener. Comput., vol. 4, no. 1, pp. 67–95, Mar. 1986.

[6] R. De Masellis, F. M. Maggi and M. Montali, "Monitoring data-aware business constraints with finite state automata", in Proceedings of the 2014 International Conference on Software and System Process - ICSSP 2014, 2014, pp. 134–143.

[7] F. M. Maggi, M. Dumas, L. García-Bañuelos, and M. Montali, "Discovering Data-Aware Declarative Process Models from Event Logs", Springer, Berlin, Heidelberg, 2013, pp. 81–96.

[8] F. M. Maggi, M. Montali, M. Westergaard, W. M. P. van der Aalst, and W. M. P. van der Aalst, "Monitoring Business Constraints with Linear Temporal Logic: An Approach Based on Colored Automata", Springer Berlin Heidelberg, 2011, pp. 132–147.

[9] F. M. Maggi, C. Di Francescomarino, M. Dumas and C. Ghidini, "Predictive Monitoring of Business Processes", Springer International Publishing, 2014, pp. 457–472.

[10] "Introducing Octavia", [Online]. Available: https://docs.openstack.org/octavia/ latest/reference/introduction.html. [Accessed: 20-May-2019].

[11] S. Sharma, D. Sahil Verma, D. Kiran Jyoti and D. Kavita, "Hybrid Bat Algorithm for Balancing Load in Cloud Computing", Int. J. Eng. Technol., vol. 7, no. 4.12, pp. 26–29, Oct. 2018.

[12] H. Rahhali, “A new conception of load balancing in cloud computing using Hybrid heuristic algorithm”, Int. J. Comput. Sci. Issues, vol. 15, no. 6, pp. 1–8, Nov. 2018.

[13] M. S. Aktas, “Hybrid cloud computing monitoring software architecture”, Concurr. Comput. Pract. Exp., vol. 30, no. 21, p. e4694, Nov. 2018.

[14] Y. C. Liu and C. L. Li, “A Stratified Monitoring Model for Hybrid Cloud”, Appl. Mech. Mater., vol. 719–720, no. Materials and Engineering Technology, pp. 900–906, Jan. 2015.

[15] K. K. Azumah, S. Kosta, and L. T. Sørenson, “Scheduling in the hybrid cloud constrained by process mining”, in Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom, 2018, vol. 2018-Decem, pp. 308–313.

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, Softw. Pract. Exp., vol. 41, no. 1, pp. 23–50, Jan. 2011.

[17] M. Montali, F. Chesani, P. Mello, and F. M. Maggi, “Towards data-aware constraints in declare”, in Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC ’13, 2013, p. 1391.

Paper D

Process mining-constrained scheduling in the hybrid cloud

Azumah, Kenneth K., Sørensen, Lene T., Montella Raffaele and Kosta Sokol

The paper has been published in the
Concurrency and Computation: Practice and Experience Vol. 33(4), pp. e6025, 2021.

The layout has been revised.

Abstract

Hybrid cloud, typically a combination of public and private cloud deployment models, is a rising paradigm due to the benefits it offers: full control of data and applications in the private cloud and elastic computing resource availability in the public cloud. This combination however brings an extra layer of complexity that can potentially erode the benefits and present serious challenges if not managed well. Among the challenges, ensuring business constraint compliance across the combination of cloud deployment models is a growing concern. Our paper brings a sensitive, data- and process-aware framework to bear on task scheduling in hybrid clouds with compliance to business constraints. Our proposed approach utilizes data from a real hybrid cloud-based hospital billing system that is governed by complex and dynamic data processing rules. Our system successfully employs a process mining controlled algorithm to schedule tasks in the hybrid cloud to comply with the given set of business constraints.

Keywords:

Hybrid cloud computing, task scheduling, process mining, event calculus, process-awareness

1 Introduction

More organizations are organizing their adoption of hybrid cloud computing technology [7, 24] in the typical collection of a private and one or more public cloud deployment models. Well-known motivations accounting for this development take advantage of having maximal control of the private portion and having an elastic supply of cloud resources from the public portion of the hybrid cloud. One such motivation is the ability to run sensitive applications without leaking their data into the public portion of the hybrid cloud. However the complexity created from the combination of the different cloud deployment models, motivates stricter monitoring for governance and regulatory compliance in order to maximize hybrid cloud adoption benefits. Also the extra layer of complexity makes task scheduling in the hybrid cloud more challenging to achieve secure and fast workload processing at minimal costs. Achieving such efficient task scheduling whilst keeping the sensitive data out the public cloud is an obvious desire for many private organizations that have adopted the hybrid cloud deployment model.

One inexpensive way of testing task scheduling algorithms is by using cloud simulators and experimenting with various configurations that match real-world conditions. CloudSim Plus[1] is one such cloud simulator written as a Java library. It is easy to modify the code to simulate various models of cloud setup in the real-world. The focus of such simulations are the efficiency of

[1] www.cloudsimplus.org

task scheduling algorithms and the enforcement of constraints placed on them. Partitioning applications and tagging sensitive data are among various methods [5, 19, 21, 26, 29, 33] proposed to address the problem of efficient scheduling in the hybrid cloud and at the same time keeping sensitive data or tasks out of the public side of the cloud. Monitoring such hybrid cloud-based applications to ensure constraints have not been violated is non-trivial, demanding a combination of interventions for success. To improve compliance to laid down business rules during data processing, the event logs of the application can be mined and analyzed to check conformance. The event log typically records the time-stamp of the event, the activity name, the resource and an identifier of the data involved in the event. Depending on the application domain, other attributes are also captured in the event log. By tracing how each additional activity is affecting the process in general, event log data can be exploited in optimizing cloud computing resources towards designated goals [9].

The task scheduler is one such programmable optimizing component of cloud resources however the extent of programmability is limited, especially in situations where the tasks are externally controlled. The programming limitations can however be overcome by monitoring event logs through process mining to augment scheduler decision-making. This form of process-mining augmented task scheduling can be applied across several domains where scheduling alone would fail to bring out the needed compliance to some specified constraints. The focus of existing works [4, 6, 10, 22] dwell on the cost and performance characteristics of scheduling in the hybrid cloud whilst the business constraint compliance aspects have seen very little research from our search of literature.

Support for business constraints is critical in hybrid cloud resource scheduling to ensure that the goals for this cloud deployment model are met, i.e. exploiting the scalability of the public cloud whilst having the desired control in the private data center. Our model achieves the satisfaction of business constraints through integration with process mining frameworks that provide a declarative mechanism for constraint specification and monitoring. It provides a mechanism of making the scheduler "data-aware" in order to influence fine-grained resource allocation and task scheduling. The scheduler examines the task and allocates VMs that will not violate the constraints.

Process mining-controlled scheduling is an option worth pursuing when executing data-intensive application in the hybrid cloud to comply with some given business constraints and meeting given QoS related deadlines. Related works in scheduling and cloud resource provisioning employ techniques such as load-awareness [32], parallelism-awareness [31], structure-awareness [18] to aid in task scheduling and middle-ware tooling [2, 15]. Data-awareness (sensitivity) but not process-awareness is considered in Oktay et al. [26]. From our best literature search efforts, no work specifically combines data sensitivity and process-awareness in task scheduling. Put together, the challenges facing business rule constrained scheduling in the hybrid cloud can be enumerated as: 1) Overcoming the complexity introduced by the combination of two or more different cloud deployment models, in this case, the typical combination of pub-

lic cloud and private cloud, on- or off-premises.
2) Checking for conformance of the task scheduling to some given business process- based constraints and reporting violations to influence further corrective task scheduling.
3) Employ specialized and pluggable middleware to provide an interface for specifying business constraints and evaluating their conformance thereof.

In this paper our contribution in respect of the above challenges can be summarized as:

- Provision of a tooling framework for task scheduling in the hybrid cloud. This we do by chaining existing tools together employing event log data. The integration of the MOBUCON[2] tool and CloudSim Plus provides a useful framework for simulation of process-aware task scheduling.
- Introduction of an algorithm to influence task scheduling to conform to the business rule of keeping sensitive tasks out the public cloud. We extend the data center broker of the CloudSim Plus simulation tool and tag the cloudlets (tasks) as either sensitive or otherwise before scheduling.
- Application of the specific scenario of a hospital billing system to test the proposed framework. We formulate the given constraints on the billing system using a declarative language and check for conformance of the scheduling to the specified constraints.

We next present in Section 2 works related to addressing hybrid cloud task scheduling with respect to some business constraints. This is followed by Section 3 containing background information on constraint specification and their monitoring, Section 4 laying out our proposed architecture and Section 5 describing the simulation setup. A real hospital scenario is presented in Section 6 and Sections 7, 8 and 9 present the results of the simulations, discussion and conclusion, respectively.

2 Related Work

Montali et al. [25] in their work introduce MOBUCON as a tool for verifying compliance to some specified business constraints. They present MOBUCON as a framework for continuous monitoring capable of integrating with existing systems via event data in the systems' logs. Implemented as a tool the description of the business constraint within is done employing the Declare Specification [27]. In their work, Declare constraints are formalized into executable forms using a logic-based framework known as Event Calculus (EC) [20], and are verified employing a logic-based "reasoner". The output of the tool shows the compliance level in the form of an overall system health measurement to provide an indication of the impact of each item in the event log reading. Put

[2] MOnitoring of BUsiness COnstraints

together, their framework illustrates how a set of complex business constraints can be specified and monitored for compliance by using the event data generated by the system.

Chesani et al. [9] adopt MOBUCON as their monitoring framework in ensuring compliance to the constraint of meeting deadlines for their MapReduce application in a hybrid cloud. Taking inspiration from Business Process Management (BPM) techniques, they specify constraints for their application. Here, the constraints describe relationships between activities that should be satisfied during process instance execution. This scenario is synonymous to the process model in the Business Process Management field, where the sequence of various activities can be said to comply or otherwise to the intended process model [30]. They show how resource provision in a hybrid cloud to meet deadlines can be influenced by the process mining monitoring properties in the MOBUCON EC [25].

SEMROD [26] presents a hybrid cloud framework for the MapReduce (MR) paradigm where sensitive data is protected from leaking into the public cloud. Their approach handles the complexity of scheduling MR intermediate keys through the public cloud such that a more than casual observer will not be able to distinguish between sensitive and non-sensitive data. SEMROD assumes an "adversary" will seek to launch statistical attacks to analyze data in order to unearth the sensitive portion of the data in transit between private and public cloud. The work does not deal conclusively with the intra-cloud bandwidth costs in the performance of the MR scheduling.

CloudSimEx [3] is part of an extension of CloudSim that explores the scheduling and cloud resource provisioning problem with a focus on comparing algorithms for efficiency. This work employs the MapReduce framework and schedules tasks towards the goal of meeting user required SLAs, mainly that of budget and deadline. Exploiting parallelism with the support of multi-core processing elements, their implementation employs a modified branch and bound algorithm to efficiently schedule MapReduce tasks on the CloudSim simulator. The authors extend CloudSim to achieve a model for simulating an efficient SLA-aware MapReduce framework for big data analytics. Their implementation is not discriminative of the location of VMs employed in the processing.

MR-CloudSim [17] presents a simulation tool for the MR paradigm by extending the CloudSim Library [8]. The motivation is to provide an inexpensive framework for simulating real-world MR applications. This they do by ensuring all Map operations are complete before the Reduce step starts. Their setup however does not deal with placement of sensitive tasks in the cloud setup.

Mattess [23] implements a cloud resource provisioning policy during task scheduling for the MR paradigm. Additional resources are made available based on the percentage of completed Map tasks per time. In their work, there is no consideration for the sensitivity of the data being processed.

Clemente-Castello et al. [11] bring to the fore trade-offs that have to be accommodated for I/O intensive scheduling in the MR paradigm. They propose locality-based scheduling incorporating rack-awareness and strategic replication to mitigate the overheads in the scheduling. Data-awareness is however not considered part of the strategy for processing.

Wang et al. [31] employ parallelism-awareness which engages available resources to complete tasks on time and achieve resource efficiency and performance. They employ algorithms in First Fit decreasing, Best Fit or Earlier Deadline First. Ghobaei-Arani et al. [15] introduce hybrid autonomic resource provisioning based on autonomic computing and machine learning techniques. Kanagaraj and Swamynathan [18] deal with under/over provisioning employing the number of tasks together with their arrangement in the workflow structure to determine VMs required for execution. The algorithms achieves more efficient utilization however does not take into consideration the sensitivity of the individual tasks. Table 1 shows the spread of related works done in task scheduling and resource provisioning in the multi- or heterogeneous cloud situation.

In this paper, we present a data-aware task-scheduling framework for the MapReduce model by utilizing process mining and the CloudSim Plus toolkit. We bring specific extensions to the toolkit to facilitate the simulation of the MapReduce model within a hybrid cloud. We present next a real-world applicative scenario based on which we will describe constraint specification frameworks employed in building our model.

3 Background to the Constraint Specification

The term constraint refers to any rule that defines a set of accepted or prohibited behaviours [25]. We apply a case of a business constraint scenario involving a hospital that has adopted a hybrid cloud for its patient billing operations. The event log of our adopted hospital billing system records the activities carried out on each patient. Process mining of such an event log can check for conformance to a defined business constraint by verifying if the sequence of the activities carried out on the patient are in the right order. Our adopted event log captures five main activities among which are relationships (or constraints) strengthening the hospital billing system. In our applicative scenario, the (business) constraints governing the Hospital Billing system are enumerated as follows: *i)*All patients must be registered before consultation. *ii)* No consultation can occur without a prior registration. *iii)* No Discharge should occur without at least one consultation. *iv)* An Episode for OPD is maximum of 168 hours. And *v)* Episode for inpatient cannot last more than 240 hours. Our paper adopts *Declare* to model these constraints and aid their formalization. The next sections, 3.1, 3.2 and 3.3 introduce the Declare language, its formalization and monitoring in respect of the constraints.

Table 1: The focus areas of related works in task scheduling and cloud resource provisioning

paper	scheduling/ provisioning technology /approach	efficient resource provisioning	avoiding SLA violations	cost minimizing	employ middleware	consider data sensitivity and process-awareness
Oktay (2015) [26]	MapReduce		✓	✓		
Chesani (2017) [9]	MapReduce, Process Mining		✓		✓	
Wang (2019) [31]		✓	✓	✓		
Kanagaraj (2018) [18]		✓	✓			
Ghobaei-Arani (2018) [15]		✓	✓	✓	✓	
Alonso-Monsalve (2018) [2]		✓			✓	
Zhang (2018) [32]		✓	✓			
Jung (2012) [17]			✓	✓		
Clemente-Castello (2016) [11]			✓	✓	✓	
Mattess (2013) [23]		✓	✓			
our work	MapReduce, Process Mining, CloudSim	✓	✓	✓	✓	✓

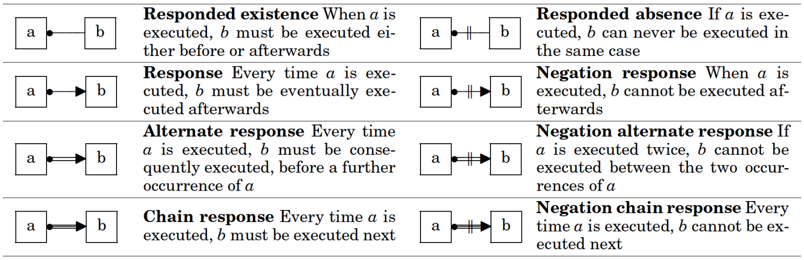


Fig. 1: Examples of commonly used Declare constraints as defined by Montali et al. [25].

3.1 The Declare Specification

Process modeling languages facilitate the visual specification of business rules. Declare is a visual modeling language that works by indicating the sequence of activities that must not be violated in executing the business process. The language has the potential for specifying constraints in hybrid cloud-based systems where an extra layer of complexity is introduced in controlling which side of the cloud the process executions can take place. The main features that make the Declare system suitable for automated operational decision support are the formal semantics-based back-end and the user-friendliness at the front-end. Figure 1 shows some Declare constraints that define rules to guide the construction of process models. The constraints are able to describe the process model of a wide array of business constraints cutting across various industries. It therefore provides a broad basis for use as a language to describe process models and business constraints. Figure 2 shows a Declare model of a simplified Hospital Billing system that has to be verified for compliance to some given constraints. The model by itself is a constraint specification using the Declare language that is composed of individual event occurrences connected in a pattern to indicate what is acceptable should events be arranged in sequence from any point in time. The figure indicates that some events cannot occur legally without some others preceding them: thus a *consultation* event cannot

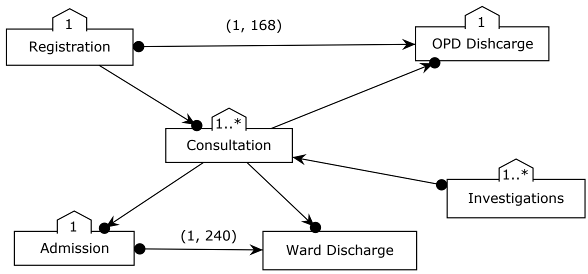


Fig. 2: The adopted Hospital Billing System expressed as a Declare constraint specification.

occur unless there is a *registration* always.

From the given rules, we designate the Discharge constraint as *con(D)* which is activated whenever there is a consultation or admission for out-patients or in-patients respectively. The *Discharge* constraint stems from the process model representing part of the given Hospital Billing system (see figure 2).The Billing constraint is adopted to keep sensitive bills out of the public portion of the hybrid cloud. In monitoring compliance to these business rules, we employ the Declare language and implement it via the *MOBUCON* framework.

3.2 Process mining monitoring

The monitoring of business constraints (MOBUCON) [25] for compliance lies at the heart of ensuring timely interventions to violations. Though it appears to be a complex endeavour to describe the intricacies of all business constraints, the availability of event logs makes information available as to the violation of constraints. The monitoring system takes as input *i*) the broad system specifications, comprising the behaviours wanted and *ii*) the acceptable order of activities as they occur in the captured event data. The process mining component interprets the specifications captured in the form of Delcare constraints and provides the feedback on the level of compliance of the trace of events. The level of compliance is an overall indicator of what proportion of the event log traces follow that acceptable sequence specified in the Declare constraint. A version of the MOBUCON tooling is based on *Event Calculus*, a logical computation framework for events, discussed further in the next section.

3.3 The Event Calculus Formalization

The Event Calculus [20] (EC) framework facilitates the formalization of constraints using a group of propositions generated from a general theory and a domain theory. Montali et al. [25] define a set of propositions better known as "predicates" in EC nomenclature that are employed in formalizing constraints. The predicates used frequently are *initially* to indicate an initial status of a constraint; *initiates* to indicate which events "activate" the constraint; *happens* to denote the occurrence of an event, in other words, establish that an event has occurred; *terminates* to indicate that the constraint has been deactivated either through the sequence of activities satisfying the constraint, or otherwise; and *holds_at* which validates an event attribute matching some variable or value. In the domain theory, predicates are leveraged in formalizing how events affect variables also known as "fluents". To monitor how the fluents are affected by events, a "reasoner" is employed to keep track of the running executions in a reactive way, incrementally calculating the validity of fluents as new events are occurring.

From Table 2, the predicates facilitate specification of the discharge constraint in our hospital billing system introduced at the beginning of section 3. The occurrence of the *regist* event for a patient ID *PID* activates the *con(D)*

Table 2: Specification of the Discharge Constraint for MOBUCON showing the status moving from pending (pend) to either violated (viol) or to satisfied (sat). Symbol abbreviations: init = initiates, comp = complete, hap = happens, term = terminates, st = status, stt = start, ex = exec, h_a = holds_at.

new instance	$init(comp(regist, PID, Out, [Pat, Hours]), st(i(PID, con(D)), pend), T) \leftarrow Hours \leq 168.$
pend to sat	$term(stt(disch, PID_2, Out_2, []), st(i(PID, con(D)), pend), T) \leftarrow h\_a(st(i(PID, con(D)), pend), T) \wedge hap(comp(regist, PID, Out, [Pat, Hours]), T_{reg}) \wedge Pat = Out_2 \wedge T > T_{reg} \wedge T \leq T_{reg} + 168.$ $init(stt(disch, PID_2, Out_2, []), st(i(PID, con(D)), sat), T) \leftarrow h\_a(st(i(PID, con(D)), pend), T) \wedge hap(comp(regist, PID, Out, [Pat, Hours]), T_{reg}) \wedge Pat = Out_2 \wedge T > T_{reg} \wedge T \leq T_{reg} + 168.$
pend to viol	$term(\_, st(i(PID, con(D)), pend), T) \leftarrow h\_a(st(i(PID, con(D)), pend), T) \wedge hap(comp(regist, PID, Out, [Pat, Hours]), T_{reg}) \wedge T > T_{reg} + 168.$ $init(\_, st(i(PID, con(D)), viol), T) \leftarrow h\_a(st(i(PID, con(D)), pend), T) \wedge hap(comp(regist, PID, Out, [Pat, Hours]), T_{reg}) \wedge T > T_{reg} + 168.$
pend to viol	$term(ex(comp), st(i(PID, con(X)), pend)).$ $init(ex(comp), st(i(PID, con(X)), viol), T) \leftarrow h\_a(st(i(PID, con(X)), pend), T).$

constraint, initiating its status to *pending.* The next events involving *PID* are monitored for the occurrence of the *disch* event within 168 hours after the initiation of the activation of the constraint. If the *disch* event for *PID* occurs after the 168 time units the constraint is deemed to have been violated and the status moves from pending pend to violated *viol.* On the other hand if the *disch* event for *PID* occurs within the required 168 time units, the *con(D)* constraint will have been satisfied moving the status from pending *pend* to satisfied *sat.*If there happens to be no occurrence of the *disch* event for *PID* and there are no more events in the event log, all pending constraints move to violated status.

Similarly, for the task scheduling stage, whenever the sequence of events for PID are Registration- Admission-Discharge a constraint for processing the bill

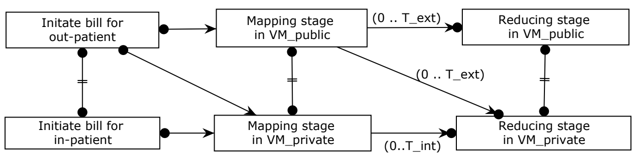


Fig. 3: Specification of constraints governing the MapReduce scheduler using the Declare visual modelling language.

in the internal data center is activated. If the billing task is carried out in the Internal Data Center (IDC) or private cloud then there is a satisfaction of the constraint otherwise there is a violation for processing in the External Data Center (EDC) or public cloud.Figure 3 shows the allowed sequence of events defining the business constraint of billing patients with the hospital system.

The MR scheduler in processing the bills has two stages Map tasks and Reduce tasks. The *MapPrivate* constraint requires that billing is done on the private cloud and it is put in the pending state when activated by the occurrence of the discharge event. When a map task is assigned to a VM_{IDC} by the scheduler, there is compliance to the constraint, moving from the pending to the satisfied state. The converse, moving from pending to violated occurs when the scheduler assigns the private map task to an off-premises (public) VM, VM_{EDC}, or delays in assigning within stipulated deadline of time T_i (see figure 3). An occurrence of the map_task_complete event instantiates a *ReduceConstraint* which moves into a satisfied state after the equivalent reduce_task event occurs and the intermediate values [12] pass into the reduce function. Any violations are regarded as task failures and trigger [12] failing MapReduce tasks to be reassigned to available VMs.

A constraint for input into the scheduler monitors the type of discharge (Ward/OPD) and schedules the map task to an appropriate VM: OPD discharge to any VM and Ward discharge to $VM'_{IDC}s$ only. The task scheduling generates traces captured in an event log. An event in the trace typically has the format *event_lifecycle, activity, case Id, resource, [Other attributes], timestamp.* The event lifecycle indicates whether the event is starting or completing and the case Id tracks the events throughout the log, building a trace. Table 2 specifies the discharge constraints in EC formalism for Declare.

3.4 Integrating the MOBUCON Framework for Constraint Specification

We link the MOBUCON tool with CloudSim Plus scheduler via event data containing traces of the sequence of activities transformed into the eXtensible Event Stream[3] (XES) format, a standard adopted by the IEEE Task Force on process mining. In relating event logs to the MOBUCON framework, event types correspond to atomic activities and life-cycle event types refers to non-atomic activities, namely: *start*, *completion* and *cancellation*, described in [28]. An example of a process execution trace in EC formalization using the *happens* predicate is $happens(ev(id, type, a), t)$, where id = event identifier, $type$ = event type, a = activity name and t = timestamp.

The concept of constraints stems from an existence of an instance or multiple instances in response to a happening [28]. Each instance can generally be represented using the notion $i(id, a, t)$ to mean a constraint identified by id, with activity a occurring at a time stamp of t. An instance of a constraint

[3]http://www.xes-standard.org/start

Table 3: Billing and SLA constraint specifications serving as an input in the MOBUCON EC tool. The specification represents the *i)*initial status of the con(bill_private) and con(sla) constraints, *ii)* status change of each constraint on account of occurrence of make_bill and bill_job events *iii)* status change of the constraint on account of an event attribute comparing to a condition. The status of each constraint changes from *pend* to *viol* or *sat* to reflect the compliance of the event data as it occurs.

initiates(complete(diagnose, PID, Out,[Patient, Sensitive]), status(i(PID, con(bill_private)), pend), T):- Sensitive = 1.
terminates(start(make_bill,PID2,Out_2,[]), status(i(PID,con(bill_private)),pend),T):- holds_at(status(i(PID,con(bill_private)),pend),T), happens(complete(diagnose,PID,Out,[Patient, Sensitive]),Ta), Patient = Out_2, T > Ta, T =< Ta + 240.
initiates(start(make_bill, PID2, Out_2,[]), status(i(PID, con(bill_private)),sat),T):- holds_at(status(i(PID, con(bill_private)), pend),T), happens(complete(diagnose,PID,Out,[Patient, Sensitive]),Ta), Patient = Out_2, T > Ta, T =< Ta + 240.
terminates(_, status(i(PID,con(bill_private)),pend),T):- holds_at(status(i(PID,con(bill_private)),pend),T), happens(complete(diagnose,PID,Out,[Patient, Sensitive]),Ta), T > Ta + 240.
initiates(_, status(i(PID,con(bill_private)),viol),T):- holds_at(status(i(PID,con(bill_private)),pend),T), happens(complete(diagnose,PID,Out,[Patient, Sensitive]),Ta),T > Ta + 240.
%% ---- MAP REDUCE SECTION------------%%
initiates(start(bill_job,PID,Out,[Server]), status(i(PID,con(sla)),pend), T):- Server = PIDc.
terminates(start(bill_job,PID,Out,[]), status(i(PID,con(sla)),pend),T):- holds_at(status(i(PID,con(sla)),pend),T), happens(start(bill_job,PID,Out,[Server]),Tj),T > Tj,T =< Tj + 10.
initiates(complete(bill_job,PID,Out,[]), status(i(PID,con(sla)),sat),T):- holds_at(status(i(PID,con(sla)),pend),T), happens(complete(bill_job,PID,Out,[Server]),Tj),T > Tj,T =< Tj + 10.
terminates(_, status(i(PID,con(sla)),pend),T):- holds_at(status(i(PID,con(sla)),pend),T), happens(start(bill_job,PID,Out,[Server]),Tj),T > Tj + 10.
initiates(_,status(i(PID,con(sla)),viol),T):- holds_at(status(i(PID,con(sla)),pend),T), happens(start(bill_job,PID,Out,[Server]),Tj), T > Tj + 10.

can have several states to represent its status in respect of the related event occurrences. The constraint states in our work use the terms "pend", "sat" and "viol" representing pending, satisfied and violated states respectively. A constraint state is *pending* if past event occurrences have instantiated it into an unfulfilled state and is waiting for the occurrence of one or more specific future events in order to move into a fulfilled state. The constraint will be *satisfied* if the specific future events with expected attributes occurred otherwise the constraint moves into the *violated* state. We therefore represent the notion of a pending instance with the term $state(i(id, a, t), pend)$ and generally denote a constraint instance in some state with *state(I,S)*, where I is constraint instance, S is current state of the constraint instance.

4 Architectural elements of the proposed framework

In our work we utilize CloudSim Plus [13], a simulation toolkit and framework that provides the environment for inexpensive experiments in task scheduling. The toolkit originates from the CloudSim [1] library, featuring an event queue for scheduling events associated with cloud entities such as data centers, virtual machines and data center brokers. The CloudSim library was re-engineered into CloudSim Plus by refactoring using SOLID and Design Pattern-based principles in software engineering [14]. The result is a cleaned-up library that is easily extendable, factors that motivated its use in this paper. Due to its extensible nature we employ the tool in our simulation of a real-world hybrid cloud scenario and embedding of process mining components.

Our proposed architecture utilizes the MOBUCON framework for the process mining monitoring aspects of the system. The framework takes as input *i)* a specification of business constraints defined in the Declare language and *ii)* formatted event log data. The output of the MOBUCON tool is an indication of how the processes mined from the event data complies with the defined business constraints. The process mining aspects serves as a classifier of workloads for allocation by the task scheduling component. For simulating an information system such as a hospital billing system, our setup employs the MapReduce paradigm for generating workloads that are governable by some specified collection of business rules. Figure 4 shows the block view of how the scheduling aspects of the CloudSim Plus toolkit is integrated with process mining support framework of the MOBUCON Client.

4.1 Utilizing the MapReduce Paradigm

We utilize the MapReduce (MR) paradigm [12] in our simulation setup to generate tasks that are governable by a collection of business constraints. MR enables the distribution of data and applications across several computers. The

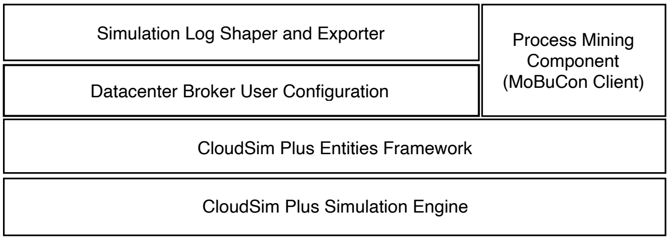


Fig. 4: Block view of the architecture showing the integration of the scheduling and process mining aspects of our proposed framework.

framework engages a cluster of computers, dividing the tasks submitted across the various nodes and processing them in parallel. In our simulations MR serves as a generator of tasks with consideration for their performance based on data content, virtual machine capacity and the location of the virtual machine node, whether in the private or public part of the hybrid cloud setup.

In our work, we extend CloudSim Plus to simulate MR principles bearing in mind the sensitivity of data being processed. This we do by enhancing CloudSim's *Cloudlet* entity class with data tagging attributes. Cloudlet is a term used by the authors of CloudSim to represent a task executed by a virtual machine. Our paper therefore uses the terms *task* and *cloudlet* interchangeably in some sections.

4.2 Running the MR Scheduler

For the purposes of the simulation, input files are packaged as a set of map tasks and expected reduce tasks. The ID's for the intermediate data that is generated from the map tasks firstly serve as keys for the data shuffling stage of the MapReduce process and secondly used for the reduce jobs. The scheduler ensures all map tasks are completed before executing the reduce tasks. This constrains the CloudSim event engine to run the MR jobs with traditional stages of map, shuffle and reduce.

The CloudSim Plus scheduler allocates tasks to VMs for processing by its event-based engine. We programme our datacenter broker to solve the scheduling problem of optimally allocating a set of tasks to a given set of machines whilst ensuring the assignation is compliant with a given constraint. The tasks in our experiment are tagged as processing sensitive or non-sensitive data and the scheduling algorithm takes note of such tasks, binding them appropriately to VMs in either the internal datacenter, IDC or public cloud, EDC, utilizing both of CloudSim's *CloudletSchedulerSpaceShared* and *CloudletScheduler-TimeShared* policy in separate experiments. The scheduler first assigns the sensitive tasks to the IDC VMs in a round-robin fashion to evenly distribute the load in IDC hosts before assigning the non-sensitive cloudlets among the remaining VMs in both IDC and EDC.

4.3 Generating MR event log in CloudSim Plus

Our logs generated from the simulated MapReduce process consists of VM creation events, the assignation of tasks (cloudlets) to the VM and the stages of executing the map and reduce tasks. Before these map and reduce stages, the goal is to assign tasks to appropriate VMs in order to satisfy the constraint of non-leakage of sensitive data in the public cloud. In logging for subsequent process mining, we apply the basic representation of an event element and its attributes. Here a *process* represents a sequence of activities that together achieve a specific goal. The order of the activities plays an important role in achieving a degree of success towards the intended goal. The more aligned the order of activities, the better the satisfaction towards the *process model*. The process model then becomes the template by which *process instances* are created and executed. One process instance leaves a *trace* of activities in the event log that together culminates in one *case* execution. The events (activities) relating to each case (process instance) can be traced in the event log and put together as a set comparable to the process model. Each event is made up of the *case id*, *event (activity) name*, *time-stamp* and *originator (resource)*. Other attributes can be added to the event to capture needed information essential to measuring compliance to the process model. In our setup we employed a commonly stored attribute that records the start and stop time of the event.

5 Setup of the Simulation and the Real Hospital Scenario

In the next sections, we present the physical environment and its configuration representing a hybrid cloud. An overview of event data generation by the MR scheduler is presented together with expected outputs and their formats. The real hospital scenario is described together with its input data and their transformations. For our environment the logical framework and integration architecture is shown in figure 5.

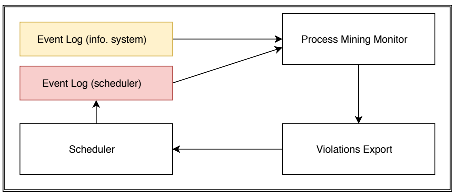


Fig. 5: High-level model of process mining monitoring architecture.

5.1 The Architectural Environment

The setup environment has 16GB RAM, 512GB disk space and a 2.5GHz quad-core processor. It has CloudSim Plus library running in IntelliJ 2018 IDE. Our setup creates two datacenters representing the public cloud or external datacenter (EDC) and the private on-premises cloud (IDC) with ten hosts each. We extend the `Cloudlet` and `Datacenterbroker` classes with data-awareness and an algorithm that facilitates matching of cloudlets to VMs with compliance to business constraints. The Datacenterbroker submits cloudlets to VMs in EDC with a delay (latency) and a bandwidth cost to model the hybrid cloud. To keep track of the consumption, bandwidths are specified for the host and datacenter to enable VMs to share total available. The cloudlets also specifies the amount of bandwidth needed for execution successfully. For cloudlets submitted to VMs in the private on-premises cloud, the relative bandwidth used is far lower due to their location on the same local network in the datacenter. The bandwidth that is exhaustible in this case is the allocated quantity on EDC. For cloudlets submitted to VMs on EDC, the bandwidth counts in determining total cost as well as sufficiency to complete their execution.

We set the hosts and VM specifications precisely to help pre-determine the exact number of VMs created during the experiments. Both datacenter hosts and VMs have four PEs of 1000 MIPS each with the cloudlet length set to 50000 MI and requiring two PEs. It is expected from this setup that processing takes 50 seconds for each cloudlet and each VM or host accommodates two cloudlets in the space-shared configuration

5.2 General Logging and Output

Our inital input data consists of two event logs made up of the event data from the hospital billing system and the event log from the scheduling system. The process mining component analyzes the combined event data from the billing system and scheduling system for compliance to the specified constraints of billing with a specified time period and processing tasks (bills) classified as sensitive in the private cloud or IDC.

Event data produced from the scheduling system serves as an input to the process mining component. For our simulation, we optimize the event data from the scheduler to the formats acceptable to the process mining tool. We use the eXtensible Event Stream (XES) format, recall the snippet shown in Table 5, for the MOBUCON tool and extract the classified data from the process mining output. Since the MOBUCON tool is able to take raw text as input, we programme the scheduler to optimize its event log for feeding into the process mining subsystem. This programming allows MOBUCON to utilize the event data without needing it to be transformed into the XES format. Once the process mining tool consumes the event data, it outputs the level of compliance to the specified business constraints using a visual feedback and an overall system health indicator to help classify the data being output.

We generate event data from the scheduling subsystem having the format *time, act, case , res, cycle, origin, dest* representing the attributes timestamp of the event; name of the activity; identity of the entity involved; the resource generating the data or the resource being used; the life-cycle indicating whether the data is the start or end of the event; the source system of the data; and the destination system of the data. On the scheduler each task assignation and execution is logged using the described format. From the entries the process mining tool is able to activate constraints such as *con(bill_private)* when a *disch* event occurs.

6 A Hospital Applicative Scenario

In this section we describe how our proposed framework helps to make decisions about the hybrid cloud configuration to employ based on the given business constraints. We first present the hospital's motivation for deploying the hybrid cloud, followed by a definition of one of its business constraints relating to data sensitivity. We show the level of data sensitivity based on constraints and process mining of real event data from its information system logs. We finally show how based on the proposed model, the hybrid cloud resources are configured to offer maximum support to business constraint conformance at minimal bandwidth costs.

6.1 The hospital and the motivation for hybrid cloud deployment

Our special applicative scenario is a hospital in Ghana with an adopted hybrid cloud deployment model. The private portion of their hybrid cloud is on-premises and the public portion is alternating between multiple cloud providers including Amazon and Microsoft Azure. The various departments of the hospital altogether attend to about 900 patients a day. The clinical bills are submitted to medical insurance partners and other sponsors at the end of each month. During batch bill processing, there is excessive load on the central application server impacting adversely on clinical operations. The hybrid cloud model is adopted to "burst" selected excess load into publicly available resources as and when needed. One militating factor in the bill processing is the lack of reliable and robust enough Internet bandwidth for cloud access during bursting of the excess load. Though the cost of the bandwidth is dropping, they remain prohibitively high and largely sporadic in nature due to deficits in infrastructure and unfavourable economic conditions to encourage investment [16]. The hospital located in this region will have to cautiously rely on the Internet for critical applications and also endeavour to make use of their internal data center to save cost in the long term.

Table 4: A section of the event data extracted and cleaned to obtain the four essential attributes of each entry: the resource (system user ID), case Id (Patient ID), activity and the timestamp.

System UserID	Patient ID	Activity	Log timestamp
...	...	...	...
nurse01	PatientID :: 012010	Admission	2017-03-08 16:39
labtech02	PatientID :: 012010	LabRequest	2017-03-08 16:42
labtech02	PatientID :: 012010	Investigation	2017-03-08 16:42
labtech02	PatientID :: 012010	Investigation	2017-03-08 16:43
labtech02	PatientID :: 012010	Investigation	2017-03-08 16:43
labtech02	PatientID :: 012010	Investigation	2017-03-08 16:43
nurse02	PatientID :: 011183	Admission	2017-03-08 20:18
labtech05	PatientID :: 011183	LabRequest	2017-03-08 20:33
labtech05	PatientID :: 011183	Investigation	2017-03-08 20:33
labtech05	PatientID :: 011183	Investigation	2017-03-08 20:33
labtech05	PatientID :: 011183	Investigation	2017-03-08 20:33
labtech05	PatientID :: 011183	Investigation	2017-03-08 20:33
pharmtech01	PatientID :: 000403	DrugSale	2017-03-08 20:38
clerk05	PatientID :: 000393	ConsultReview	2017-03-09 10:43
clerk05	PatientID :: 000393	TreatCharges	2017-03-09 10:43
clerk10	PatientID :: 008452	Admission	2017-03-09 12:32
clerk11	PatientID :: 007926	ConsultDoctor	2017-03-10 4:22
labtech03	PatientID :: 007926	LabRequest	2017-03-10 5:00
labtech03	PatientID :: 007926	Investigation	2017-03-10 5:00
labtech03	PatientID :: 007926	Investigation	2017-03-10 5:00
labtech03	PatientID :: 007926	Investigation	2017-03-10 5:04
doctor03	PatientID :: 007926	Admission	2017-03-10 6:06
doctor03	PatientID :: 007926	ConsultReview	2017-03-10 6:07
doctor03	PatientID :: 007926	Diagnosis	2017-03-10 6:07
doctor03	PatientID :: 007926	Diagnosis	2017-03-10 6:07
clerk13	PatientID :: 029437	Visitation	2017-03-11 12:42
clerk13	PatientID :: 029437	ConsultDoctor	2017-03-11 12:42
clerk01	PatientID :: 022909	Admission	2017-03-11 12:45
passist01	PatientID :: 029437	ConsultReview	2017-03-11 12:48
passist01	PatientID :: 029437	Diagnosis	2017-03-11 12:48
clerk13	PatientID :: 029437	Visitation	2017-03-11 12:48
clerk13	PatientID :: 029437	ConsultDoctor	2017-03-11 12:48
passist01	PatientID :: 000400	ConsultReview	2017-03-11 19:18
passist01	PatientID :: 000400	Diagnosis	2017-03-11 19:18
clerk14	PatientID :: 000100	Visitation	2017-03-11 19:57
clerk14	PatientID :: 000100	ConsultDoctor	2017-03-11 19:57
labtech02	PatientID :: 000100	LabRequest	2017-03-11 20:26
labtech02	PatientID :: 000100	Investigation	2017-03-11 20:26
labtech02	PatientID :: 000100	LabRequest	2017-03-11 20:28
labtech02	PatientID :: 000100	Investigation	2017-03-11 20:28
labtech02	PatientID :: 000100	Investigation	2017-03-11 20:28
clerk03	PatientID :: 029437	LabRequest	2017-03-12 13:28
clerk03	PatientID :: 029437	Investigation	2017-03-12 13:28
clerk03	PatientID :: 029437	Investigation	2017-03-12 13:28
clerk01	PatientID :: 029437	ConsultReview	2017-03-12 13:29
clerk01	PatientID :: 029437	TreatCharges	2017-03-12 13:29
clerk01	PatientID :: 029437	Admission	2017-03-12 13:30
clerk12	PatientID :: 012010	LabRequest	2017-03-12 13:37
...	...	...	...

Table 5: A snippet of the transformed event log in the XES format.

```
<string key="concept:name" value="PatientID :: 000100"/>
<event>
<string key="concept:instance" value="0"/>
<string key="System UserID" value="clerk14"/>
<string key="concept:name" value="Visitation"/>
<string key="lifecycle:transition" value="start"/>
<date key="time:timestamp" value="2017-03-11T19:57:00.000Z"/>
</event>
<event>
<string key="concept:instance" value="0"/>
<string key="System UserID" value="clerk14"/>
<string key="concept:name" value="Visitation"/>
<string key="lifecycle:transition" value="complete"/>
<date key="time:timestamp" value="2017-03-11T19:57:00.000Z"/>
</event>
<event>
<string key="concept:instance" value="1"/>
<string key="System UserID" value="clerk14"/>
<string key="concept:name" value="ConsultDoctor"/>
<string key="lifecycle:transition" value="start"/>
<date key="time:timestamp" value="2017-03-11T19:57:00.000Z"/>
</event>
<event>
<string key="concept:instance" value="1"/>
<string key="System UserID" value="clerk14"/>
<string key="concept:name" value="ConsultDoctor"/>
<string key="lifecycle:transition" value="complete"/>
<date key="time:timestamp" value="2017-03-11T19:57:00.000Z"/>
</event>
<event>
<string key="concept:instance" value="2"/>
<string key="System UserID" value="labtech02"/>
<string key="concept:name" value="LabRequest"/>
<string key="lifecycle:transition" value="start"/>
<date key="time:timestamp" value="2017-03-11T20:26:00.000Z"/>
</event>
```

6.2 Specification of the selected business constraint of the hospital

Clinical activities revolve mainly around two categories: in-patients where patients are admitted to a hospital ward for treatment usually spanning more than one day; and out-patients where the patient is treated within the day in some consulting room. The hospital designates the in-patient clinical pathways as producing sensitive data. To determine if a given event data set has the sequence of activities culminating in sensitive data, we define the business constraint in the Declare specification language. For this scenario activities after an "Admission" event are deemed as generating sensitive data which warrants the processing the admitted patient's bill in the internal data center. Figure 6 shows the representation of the business constraint using the Declare specification language.

We initially extract a section of event data from the information system logs and clean it to focus on the four essential attributes necessary for process mining: resource, case Id, activity, timestamp. A snippet is shown in Table 4. Our section of event data extracted which covers four days in March 2017, has 6471 cases (patient IDs) and 194622 events occurrences, is transformed to XES[4] format (shown in Table 5) as input for the process mining monitor. Relating Figure 6 to Table 6 *Admission* and *TreatCharges* occur 1032 and 13520 times respectively giving an indication of the percentage sensitive data (patient bills) to be processed. Since it is rare that a patient is admitted twice in an episode of hospital attendance, the percentage sensitive data is made equivalent to the occurrences of the Admission event and calculated as follows:

$$percentage\ sensitive\ data = \frac{number\ of\ inpatients}{total\ number\ of\ patients} \Rightarrow \frac{1032}{6471} \approx 16\% \quad (1)$$

6.3 Configuration of the hospital's hybrid cloud for optimal business constraint compliance

From Equation 1, the hybrid cloud can be configured to optimally accommodate the business constraint of keeping sensitive tasks in the private data center during cloud "bursting". It is also clear that the sensitive tasks are more data-intensive on account of the large percentage of activities related to cases in the in-patient category. Since these sensitive tasks are constrained to be processed

[4] http://xes-standard.org

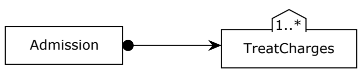


Fig. 6: The Declare representation of the business constraint where "Admissions" must eventually be followed by one or more "TreatCharges".

Table 6: An analytical view of the event log extracted from the information system, showing frequency of occurrence of activities in absolute count and in proportion.

Activity	**Occurrences (absolute)**	**Occurrences (relative)**
DrugSaleItems	31448	16.159%
Diagnosis	31374	16.12%
Investigation	25008	12.85%
ConsultReview	21378	10.984%
ConsultDoctor	18676	9.596%
Visitation	18666	9.591%
TreatCharges	13520	6.947%
DrugSale	11210	5.76%
Receipt	8398	4.315%
LabRequest	7590	3.9%
Patient	3032	1.558%
InsuredPatient	2522	1.296%
Admission	1032	0.53%
DrugReturnItems	284	0.146%
VisitationPro	156	0.08%
DrugReturn	134	0.069%
ReceiptPro	128	0.066%
PatientCredit	30	0.015%
PatientWaiverItems	12	0.006%
PatientWaiver	12	0.006%
PatientRefund	12	0.006%

in the internal data center by our scheduling algorithm, there is an average 25% bandwidth cost reduction over the unconstrained algorithm. This is obtained from the total amount of data attributable to in-patients' processing: each patient having an average of 25 activities (units of billing data). From Table 6, the occurrence of *TreatCharges* events are 13 times that of *Admission* events. Added to other essential activities conducted for an admitted patient, each in-patient has an average of 25 data units culminating in bandwidth savings if they are processed in the internal data center. It is noteworthy that in this particular scenario, adjusting the number of external VMs for the hospital has not impact on the level of violations experienced.

The next section shows the effect of varying the percentage of sensitive tasks and external VMs on the overall processing time and bandwidth costs.

7 Simulation Results

Our experiments compare performance characteristics of the hybrid cloud against an on-premises-cloud only situation. We vary the composition of the hybrid cloud with IDC:EDC ratio ranging from 10:10 through to ratio 10:1 for hybrid cloud and finally to 10:0 for an all-private cloud setup. For the purpose of demonstrating the effect from the process mining, we also vary the proportion of sensitive tasks for scheduling from 5% through 100%.

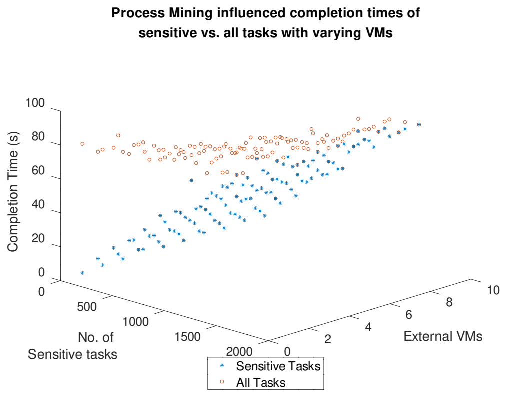


Fig. 7: Comparing the completion times of the sensitive tasks with that of all tasks under the process mining influenced algorithm: the times for the sensitive tasks increase as their proportion grows, with minimal impact by the number of external VMs. The times for all tasks together remain relatively constant with a slight dip around 1000 sensitive tasks and 4 external VMs under the process mining-influenced scheduling algorithm.

7.1 Base performance and effect of increasing EDC virtual machines

In the simulations using our process mining-influenced algorithm, we observe the effect of increasing external VMs on the completion time of sensitive tasks on one hand and all tasks on the other. The times for the sensitive tasks increases with increasing proportion of sensitive tasks as seen in Figure 7. The rising completion times trend for the sensitive tasks however do not signif-

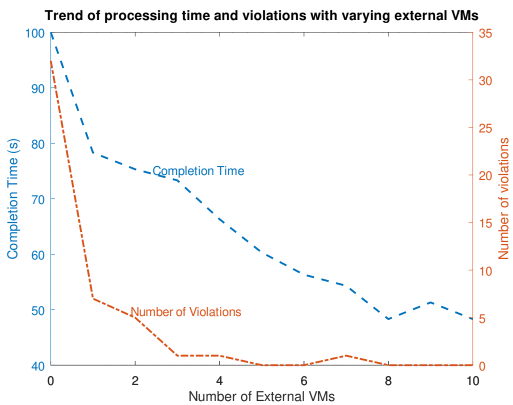


Fig. 8: Decreasing violations and completion times as external VMs increase. This base scenario employs the naive round-robin scheduling that does not take into account the sensitivity of the tasks.

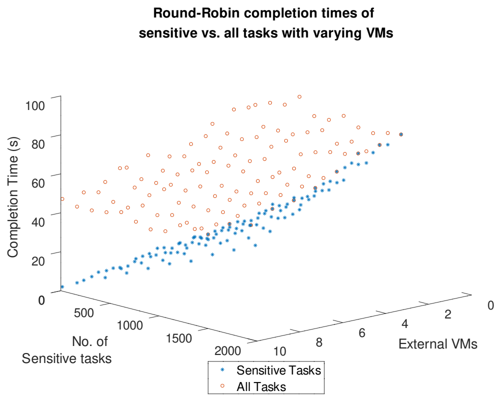


Fig. 9: Under the unconstrained round-robin algorithm, the completion times of the sensitive tasks increases with increasing proportion of sensitive tasks. For all tasks the completion times decreases slightly as the number of external VMs increases.

icantly change on account of increasing the number of external VMs. The completion times trend for all tasks however changes direction affected by both the proportion of sensitive tasks and the number of external VMs. Thus from Figure 7, the times for all tasks together remain relatively constant with a slight dip around 1000 sensitive tasks and 4 external VMs under the process mining-influenced scheduling algorithm.

For the base scenario where the naive Round-Robin algorithm is applied

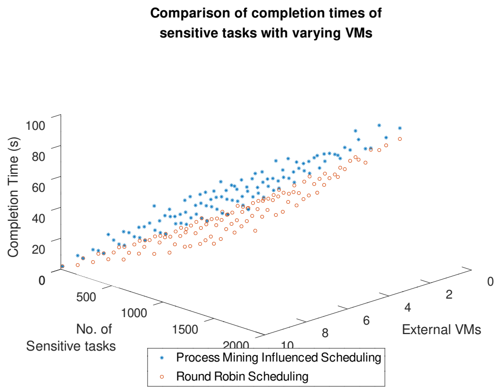


Fig. 10: The completion times of sensitive tasks increase with increasing number of sensitive tasks under the process mining-influenced scheduling with marginal impact from the number of external VMs. The completion time of the sensitive tasks are slightly faster under the round-robin algorithm for most proportions of sensitive tasks. The impact of the external VMs on the completion times is significant under the unconstrained round-robin algorithm.

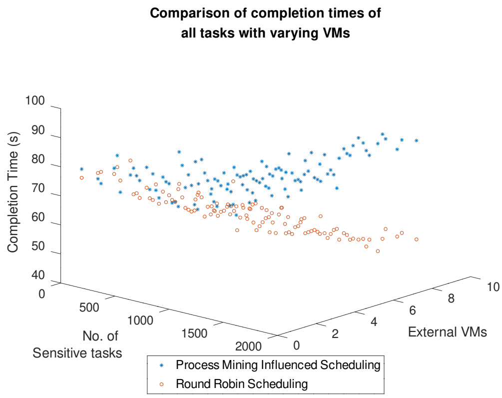


Fig. 11: The completion times of all tasks under the process mining-influenced algorithm compared to the completion time under the unconstrained Round-Robin algorithm. Completion times decrease under the unconstrained algorithm with increasing number of external VMs but increase sharply at about 1000 sensitive tasks under the process-mining influenced algorithm with minimal impact from the number of external VMs.

without the constraint of task sensitivity, we observe an expected decrease in processing time of all tasks as the number of EDC VMs increase from one through 10, as depicted in Figure 8. Our all-private cloud scenario establishes a baseline of the simulations indicating close to 100% compliance with minimal violations from the process mining monitor. This is mostly stemming from the fact that all tasks are executed inside the IDC but with some possibility of failure to meet deadlines in situations where the number of tasks *exceed* IDC capacity. Thus increasing the percentage of sensitive tasks raises the completion time which also decreases marginally for all tasks as more EDC VMs are added to the setup. We prioritize the compliance aspects over the cost for processing the hybrid cloud to ensure as much as possible sensitive data is protected. We observe the overall processing takes longer as the proportion of sensitive tasks increases for any fixed quantum of external VM resource. Our secondary objective after compliance is to reduce the overall completion time by spawning more EDC VMs to process non-sensitive tasks. From Figure 8, increasing the external VMs decreases both the overall processing time and the number of violations. From the figure, we estimate optimal quantum of EDC resource that should be allocated per percentage of sensitive data from the process mining monitoring.

In Figure 9, under the unconstrained algorithm, the completion time of all tasks decreases as the external VMs increases. This is owing to the fact that the sensitive tasks are not constrained to the IDC VMs only and since the total number of tasks remains the same, at 1500, the completion times depend solely on the number of external VMs. The completion times of the proportion of sensitive tasks increases even with increases external VMs. This indicates the

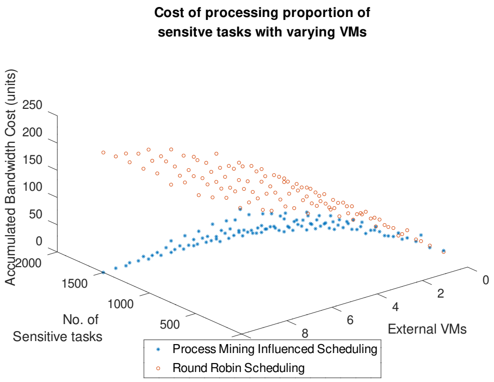


Fig. 12: Comparison of accumulated bandwidth of constrained vs. unconstrained task scheduling. Under the constrained algorithm, more of the sensitive tasks are processed in the internal data center resulting in lower bandwidth costs. The unconstrained algorithm however sacrifices cost in order to meet deadlines.

normal processing of tasks in all available VMs. Directly comparing sensitive tasks to all tasks under the constrained scheduling algorithm in Figure 10, the completion time rises with the proportion of sensitive tasks irrespective of the number of external VMs. Thus the number of external VMs have no impact on the completion times of sensitive tasks validating the restrictions imposed by the scheduling algorithm in the assignation. The completion time for all tasks remains in range of the average with a slight dip between four and six EDC VMs and at about 50% sensitive tasks. The time begins rising again as the sensitive tasks proportion begins increasing from the 50% neighbourhood and beyond the six EDC VMs

Figure 11 shows the comparison of completion of the sensitive tasks under the constrained algorithm on one hand and the non-constrained algorithm on the other. From the figure, the constrained algorithm produces later completion times than the unconstrained scheduling of the sensitive tasks. Once again, this effect happens in the overall time for all tasks, having a larger completion time under the constrained algorithm in Figure 11: the difference in the completion time increases with increasing external VM. Among the sensitive cloudlets completion time however, there is an increase as the proportion of sensitive tasks increases.

7.2 Cost aspects of optimizing the hybrid cloud

We measured accumulated bandwidth on all tasks executed both with and without consideration for sensitivity constraints. Under the algorithm taking into consideration task sensitivity, the higher the percentage of sensitive tasks the lower the cost per task, holding quantity of EDC VMs constant. Figure 12 shows a decreasing accumulated bandwidth cost as the proportion of sensitive tasks increase for each allocated quantity of external VMs. This shows the effect of the constraining algorithm to keep sensitive tasks out of the external VMs at the expense of possible deadline constraint violations.

7.3 Compliance aspects of optimizing the hybrid cloud

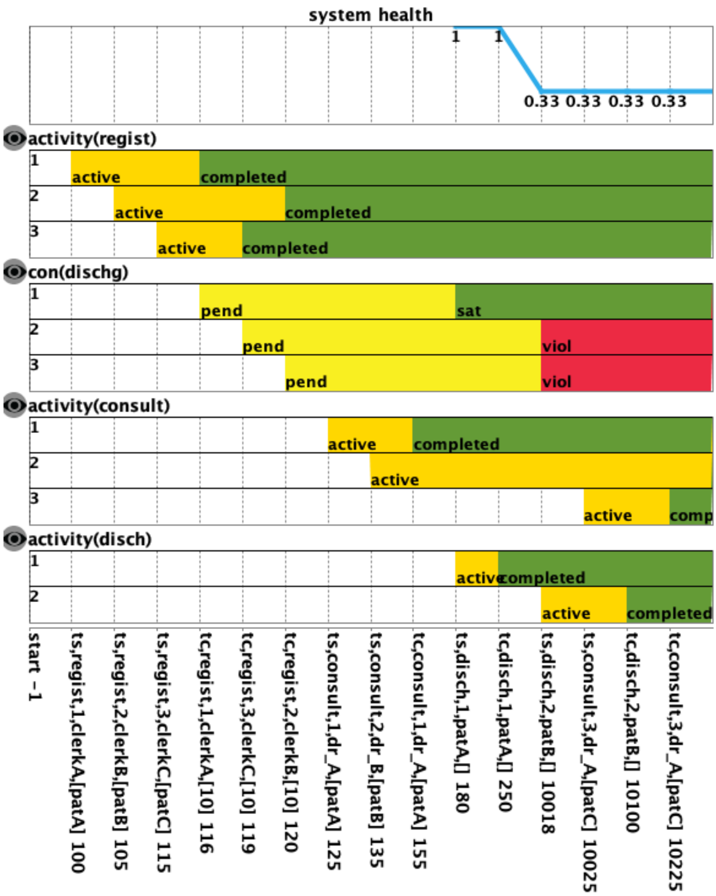


Fig. 13: Output of process mining monitoring of the Discharge constraint showing how the proportion of violations per time impacts on the overall compliance to the specified constraints.

Optimal configurations influenced by process mining is made possible through monitoring at the user friendly interface of the MOBUCON EC tool. Figure 13 displays the result of applying our running Discharge constraint on a section of event data through the MOBUCON tool. The output is characteristic from the MOBUCON monitoring framework depicting the compliance level of the sequence of activities in our selected section of event log and how the occurrence of each event impacts on the overall compliance level. The figure also shows relationship between the occurrence of the events at specific units of time and the system health, an indication of the compliance level to the constraints. The output tracks the compliance to the constraint placed on out-patient operations. The Discharge constraint con(D) is activated when a *regist* event occurs, starting and ending T_{reg_a} at timestamps 100 and 116 respectively. The activation of the constraint sets its status to *pend* waiting for the occurrence of a *disch* event for patient *patA* by timestamp $T_{ext} = T_{reg\_a} + 168$ in order to comply with the business constraint. The *disch* event for *patA* occurs at timestamp $T_{dis\_a} = 180$ satisfying the pending constraint ($T_{dis\_a} \leq T_{ext} = T_{reg\_a} + 168$) and therefore setting its status to *sat*. At time unit 180, the discharge constraint is satisfied by the occurrence of *disch* event, resulting in a change of color and status to *sat*. Beyond T_{ext}, the occurrence of the *disch* event for *patB* results in a violation since $T_{dis\_b} = 10018$ and $T_{dis\_b} > T_{ext} = T_{reg\_b} + 168$.

7.4 Visual output of system health during process mining monitoring

Figure 14 depicts the output of process mining monitoring activity on both the business constraints of *i)* restricting in-patient bills to the private cloud or IDC and *ii)* processing the bill with the time frame specified by the SLA. As the *diagnosis* event occurs, the *con(bill_private)* constraint is activated for the patient ID involved. Our setup puts *con(bill_private)* in the pending state awaiting the occurrence of a *make_bill* event.

From the afore-mentioned figure the system log captures the completion of diagnosis on in-patient *patA* at time mark 116. Constraint *con(bill_private)* is activated and put in the pending state at the same time mark. The next event related to *patA* is the start of *make_bill* event at time mark 180 and completion at time mark 350. Since processing of the bill started at time mark 180 and $180 - 116 = 64 < 240$, there is a fulfilment of the constraint to bill the in-patient within the required 240 hours (10 days) as specified in Table 3. The next event to occur is the *diagnosis* event for in-patient *patD* at time mark 400. At this point, the pending *con(bill_private)* constraints for in-patients *patB* and *patC* are awaiting respective *make_bill* events within the stipulated 240 time units in order to fulfil the constraint. The completion time marks of the *diagnosis* events for both *patB* and *patC*, 120 and 119, respectively, make their constraints violated since $400 - 120 > 240$, the max allowed time units for initiating an inpatient bill is exceeded. The color of the violated constraint changes from the time mark 400 and onward.

The figure provides a visual overview of the state health of the scheduling system which indicates the proportion of compliance to all constraints activated. At time mark 400 the system health is 0.33 because one out of three constraints has been satisfied. At time mark 600, the system health improves to 0.5 because two out of four constraints have been satisfied. Hence by mon-

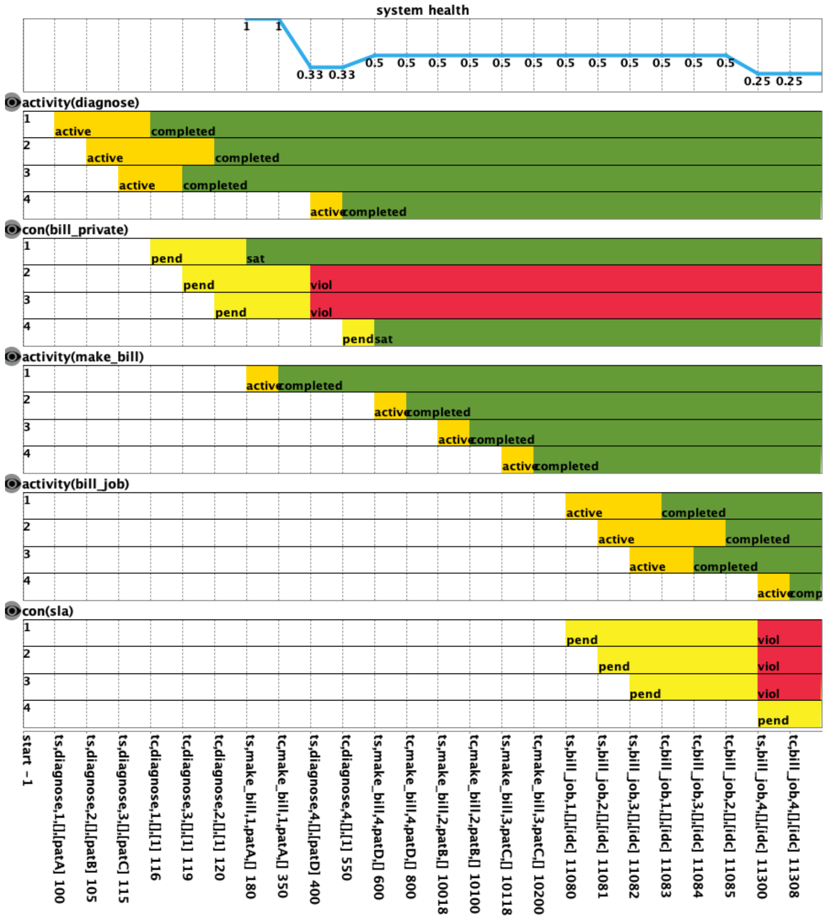


Fig. 14: Output of process mining monitoring of the Discharge constraint showing how the proportion of violations per time impacts on the overall compliance to the specified constraints.

itoring the overall system health, users can determine if the hybrid cloud is optimally structured to accommodate the business constraints.

8 Discussion of Results

The amount of external cloud resource needed to schedule tasks according to a given constraint largely depends on the quantum of tasks and the scheduling constraints. Generally, the more the cloud resource the less constrained the processing time; and the more the tasks the greater the completion time with a fixed cloud resource. Thus completion time of processing tasks with the constraint of assignation only to limited portion of a hybrid cloud increases as the number of tasks increases. Our results and scheduling algorithm generally follow these constraints and provide an indication of the optimal level of cloud resources to engage in order to comply with a given set of business constraints.

We have obtained results from a real hospital billing system used as our applicative scenario. The hospital's hybrid cloud-based billing system is governed by a collection of business constraints which changes to accommodate necessary alterations to the clinical pathways of the organization. Because of unpredictable nature of healthcare treatment processes, there is a heavy reliance on event logs to determine if the clinical processes are in compliance with the laid out constraints. In our scenario, the scheduler in assigning bill processing tasks to the hybrid cloud VMs makes a decision based on the classification of the bill by the underlying process mining controlled algorithm. From the results in Figures 7, 9, 10, 11 and 12, our scheduling algorithm prioritizes compliance over speed because of the unique nature of the requirements or business constraints. Thus it chooses to process the bill in the IDC on account of privacy, regulatory requirement and cost concerns. This means the aim of avoiding processing in the public cloud is achieved, bandwidth besides avoiding excessive reliance on possibly unreliable regional Internet bandwidth. Comparatively, satisfying business constraints in a hybrid cloud calls for trade-offs in the direction of where the organization will draw the most benefits. In our hospital scenario, located in a region that has relatively high Internet bandwidth costs, our process mining controlled algorithm is able to filter out data intensive tasks for processing in the internal data center or the on-premises private cloud, thereby saving on bandwidth costs.

Tool-chaining the process mining and task scheduling components makes our solution modular and highly flexible. This is because of our unique method of integrating the components via the event logs and their transformed derivatives. This means once a component is able to consume as input, event log data in some form, it can function as part of the system to optimize scheduling in the hybrid cloud.

Some limitations of our approach is the prioritizing of process-aware compliance at the expense of the optimal speed of task performance. However it is commonly understood that within a hybrid cloud, there are usually some form of trade-offs. Also the benefits of our approach include the inexpensive framework for determining the configuration of hybrid cloud that optimally complies with organizational complex and dynamic data processing constraints.

8.1 Summary of Findings

We deduced two main benefits from the work. Firstly, the difference in sensitive task completion times between the process mining-influenced algorithm and the Round-Robin scheduling is minimal with the maximum of 40% and an average of 10% at 100% proportion of sensitive tasks with maximum number of external VMs. This implies during scheduling when all tasks are sensitive, the process mining influenced algorithm experienced an average of 10% slower time than the round-robin algorithm. This is an acceptable result considering that there is full compliance to the given business rule of keeping sensitive tasks in the private portion of the hybrid cloud.

Secondly, by varying the continuum of sensitive tasks in our simulations, we discover the right balance between the proportion of sensitive tasks, the number of external VMs and amount of on premises data center resource to engage in order to obtain a desired completion time and other business rule compliance. This facilitates data center resource planning in our business scenario of a hospital where it is challenging to estimate the proportion of incoming sensitive tasks ahead of time.

8.2 Application in other Domains

In this section, we present a brief overview of how our framework applies to other domains aside the healthcare sector. We select the financial services sector as it is one of the avid adopters of cloud technologies. As the hybrid cloud is becoming increasing important in keeping pace with security, regulatory compliance and data privacy concerns in the financial services sector, the matter of increased complexity, limited visibility and control also gain traction. In order to help address these issues from our tooling framework perspective, the financial services sector also engages the concept of business constraints to gain a handle of specified aspects of monitoring. Taking the scenario of adoption of a hybrid cloud in order to participate in open banking services, where third-party providers (TPPs) create value added services on top of traditional banking, an existing bank would need to monitor the processes occurring on the public side of the hybrid cloud to ensure maximum regulatory compliance, especially in the processing of sensitive data. The bank can tag the records of various categories of clients in order for scheduling algorithms to process in a data-aware fashion. The Declare language allows further business constraints to be specified on the processing of such data. Our algorithm takes into account the violations from the process mining monitoring based on the specified constraints and employs it in scheduling further tasks appropriately for compliant processing. Our tooling framework enables violations to be logged and displayed for remedial actions including adjusting the resources in hybrid cloud in order to reduce the violations. The event log generated by APIs connecting various TPPs becomes the fundamental monitoring integrator via process mining thereby providing a higher level visibility and control in the hybrid cloud.

The scheduling algorithm's ability to incorporate the process mining monitoring output provides a broad foundation for applicable use in a wide variety of service-based industries, especially when the business constraints governing the industry are specified in the Declare format.

9 Conclusion

This paper explores an issue of task scheduling governed by a given set of business rules in a hybrid cloud. It investigates how the performance of tasks are impacted by the given constraints and the level of cloud resource required to bring the performance and cost to near-optimal configuration. Business rules change quite often in response to conditions external to an organization. This mostly translates to a modification of information systems for compliance purposes. A hybrid cloud introduces an extra layer of complexity owing to the combination of private and public deployment models and therefore establishes the problem of monitoring task processing for compliance to some constraints. Our proposed model integrates the CloudSim Plus scheduler over a MapReduce paradigm and MOBUCON EC tool to produce a cost-effective architecture for monitoring scheduling of tasks for conformance to business rules. Our simulations employs an initial event log of a hospital information system as input for process mining and produces as output a set of tasks tagged as either sensitive or otherwise. The scheduling of the tagged tasks are in turn monitored for compliance to the constraint of assigning only to the private portion of the hybrid cloud and processing with stipulated deadlines. Our results validate the functioning of a process mining controlled scheduling algorithm and gives an indication of the level of cloud resources needed to optimally comply to changing constraints. The MapReduce paradigm is employed in demonstrating how information system applications can be monitored for compliance to business constraints within the hybrid cloud. The simulation setup exploits the ease of use of CloudSim Plus and the out of the box functionality of the MOBUCON EC framework. Our future work extend task scheduling with process mining monitoring from simulated systems to physical systems such as OpenStack[5] to explore effects of process mining monitoring on costs and efficiency. It will also attempt to explore mathematical models for cloud resource allocation per process mining output.

[5]https://www.openstack.org/

References

[1] *CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services*, ser. arXiv, 0903.2525, mar 2009.

[2] S. Alonso-Monsalve, F. García-Carballeira, and A. Calderón, "A heterogeneous mobile cloud computing model for hybrid clouds," *Future Generation Computer Systems*, vol. 87, pp. 651–666, 2018.

[3] M. Alrokayan, A. V. Dastjerdi, and R. Buyya, "SLA-Aware Provisioning and Scheduling of Cloud Resources for Big Data Analytics," in *2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, ser. 2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2014, pp. 1–8.

[4] S. Atiewi, S. Yussof, and M. Ezanee, "A Comparative Analysis of Task Scheduling Algorithms of Virtual Machines in Cloud Environment," *Journal of Computer Science*, vol. 11, no. 6, pp. 804–812, 2015.

[5] K. K. Azumah, S. Kosta, and L. T. Sørenson, "Scheduling in the hybrid cloud constrained by process mining," in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, ser. Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, dec 2018, pp. 308–313.

[6] Y. Balagoni and R. R. Rao, "A cost-effective SLA-aware scheduling for hybrid cloud environment," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, ser. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2016, pp. 1–7.

[7] C. Baun and M. Kunze, "A Taxonomy Study on Cloud Computing Systems and Technologies," in *Cloud Computing*, ser. Cloud Computing. CRC Press, oct 2011, pp. 73–90.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, jan 2011.

[9] F. Chesani, A. Ciampolini, D. Loreti, and P. Mello, "Map reduce autoscaling over the cloud with process mining monitoring," in *Communications in Computer and Information Science*, ser. Communications in Computer and Information Science. Springer, Cham, apr 2017, pp. 109–130.

[10] N. Chopra and S. Singh, "Deadline and cost based workflow scheduling in hybrid cloud," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, ser. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 840–846.

[11] F. J. Clemente-Castelló, B. Nicolae, R. Mayo, J. C. Fernández, and M. M. Rafique, "On exploiting data locality for iterative mapreduce applications in hybrid clouds," in *Proceedings of the 3rd IEEE/ACM International Conference on*

Big Data Computing, Applications and Technologies - BDCAT '16, ser. Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT '16. New York, New York, USA: ACM Press, dec 2016, pp. 118–122.

[12] J. Dean and S. Ghemawat, "MapReduce," *Communications of the ACM*, vol. 51, no. 1, p. 107, jan 2008.

[13] M. C. Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio, and M. M. Freire, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," in *Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management*, ser. Proceedings of the IM 2017 - 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management, 2017, pp. 400–406.

[14] M. Fowler, *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 2002.

[15] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, "An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach," *Future Generation Computer Systems*, vol. 78, pp. 191–210, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.02.022

[16] International Telecommunication Union (ITU), "Study on international internet connectivity in sub-saharan africa," https://www.itu-ilibrary.org/content/publication/pub-80981022-en, p. 55, 2013. [Online]. Available: https://www.itu-ilibrary.org/content/publication/pub-80981022-en

[17] J. Jung and H. Kim, "MR-CloudSim: Designing and implementing MapReduce computing model on CloudSim," in *2012 International Conference on ICT Convergence (ICTC)*, ser. 2012 International Conference on ICT Convergence (ICTC). IEEE, oct 2012, pp. 504–509.

[18] K. Kanagaraj and S. Swamynathan, "Structure aware resource estimation for effective scheduling and execution of data intensive workflows in cloud," *Future Generation Computer Systems*, vol. 79, pp. 878–891, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.09.001

[19] N. Kaviani, E. Wohlstadter, and R. Lea, "Partitioning of web applications for hybrid cloud deployment," *Journal of Internet Services and Applications*, vol. 5, no. 1, p. 14, dec 2014.

[20] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, pp. 67–95, mar 1986.

[21] G. Levitin, L. Xing, and Y. Dai, "Optimal data partitioning in cloud computing system with random server assignment," *Future Generation Computer Systems*, vol. 70, pp. 17–25, may 2017.

[22] X. Mao, C. Li, W. Yan, and S. Du, "Optimal Scheduling Algorithm of MapReduce Tasks Based on QoS in the Hybrid Cloud," in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, ser. 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2016, pp. 119–124.

[23] M. Mattess, R. N. Calheiros, and R. Buyya, “Scaling MapReduce Applications Across Hybrid Clouds to Meet Soft Deadlines,” in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, ser. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE, mar 2013, pp. 629–636.

[24] P. Mell and T. Grance, “Effectively and Securely Using the Cloud Computing Paradigm,” *NIST, Information Technology Laboratory*, 2009.

[25] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. van der Aalst, “Monitoring business constraints with the event calculus,” *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 1, pp. 1–30, dec 2013.

[26] K. Y. Oktay, S. Mehrotra, V. Khadilkar, and M. Kantarcioglu, “SEMROD: Secure and Efficient MapReduce Over HybriD Clouds,” in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD ’15. New York, NY, USA: ACM, 2015, pp. 153–166.

[27] M. Pesic and W. M. P. van der Aalst, “A Declarative Approach for Flexible Business Processes Management,” in *Business Process Management*, ser. Business Process Management. Springer-Verlag Berlin, 2006, pp. 169–180.

[28] M. Pesic, H. Schonenberg, and W. M. van der Aalst, “DECLARE: Full Support for Loosely-Structured Processes,” in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, ser. 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007). IEEE, oct 2007, pp. 287–287.

[29] M. Smit, M. Shtern, B. Simmons, and M. Litoiu, “Partitioning applications for hybrid and federated clouds,” *Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research*, pp. 27–41, 2012.

[30] W. Van der Aalst, J. Desel, and A. Oberweis, *Business Process Management: Models, Techniques, and Empirical Studies.* Springer, 2000, vol. 1806.

[31] B. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, “Improving task scheduling with parallelism awareness in heterogeneous computational environments,” *Future Generation Computer Systems*, vol. 94, pp. 419–429, 2019.

[32] F. Zhang, J. Ge, Z. Li, C. Li, C. Wong, L. Kong, B. Luo, and V. Chang, “A load-aware resource allocation and task scheduling for the emerging cloudlet system,” *Future Generation Computer Systems*, vol. 87, pp. 438–456, 2018.

[33] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, “Sedic: Privacy-aware Data Intensive Computing on Hybrid Clouds,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS ’11). New York, NY, USA: ACM, 2011, pp. 515–526.

Paper E

Modelling and Simulating a Process Mining-Influenced Load-Balancer for the Hybrid Cloud

Kenneth Kwame Azumah, Paulo Romero Martins Maciel, Lene Tolstrup Sørensen, Sokol Kosta

The paper has been accepted for publication in the
IEEE Transactions in Cloud Computing DOI 10.1109/TCC.2022.3177668.

Abstract

The hybrid cloud inherits the best aspects of both the public and private clouds. One such benefit is maintaining control of data processing in a private cloud whilst having nearly elastic resource availability in the public cloud. However, the public and private cloud combination introduces complexities such as incompatible security and control mechanisms, among others. The result is a reduced consistency of data processing and control policies in the different cloud deployment models. Cloud load-balancing is one control mechanism for routing applications to appropriate processing servers in compliance with the policies of the adopting organization. This paper presents a process-mining influenced load-balancer for routing applications and data according to dynamically defined business rules. We use a high-level Colored Petri Net (CPN) to derive a model for the process mining-influenced load-balancer and validate the model employing live data from a selected hospital.

Keywords:

Hybrid Cloud, process mining, load balancing, event monitoring, Colored Petri Net

1 Introduction

Cloud computing has emerged as a solution for Internet-delivered computing-as-a-utility that can serve many spheres of industry. The hybrid cloud computing platform, a combination of public cloud and private cloud infrastructures, addresses some particular needs of adopters such as controlling sensitive data processing whilst enjoying near-elastic resource availability. Because of advantages offered by the hybrid cloud [1–3], many organizations are adopting a cloud deployment model that allows for easy and rapid scalability, secures control over sensitive business data, and also enables privacy in processing such sensitive data[1]. The compound of the public and private cloud, however, introduces some challenges [2–7]. Among them, we may stress the differences in the level of access and control of their underlying infrastructure: there is total control in an on-premises private cloud whilst the level of control over the provisioned public cloud resource is relatively limited depending on the service model. With increasing complexity and ever-changing characteristics of business constraints [8], the higher the control (visibility) over the hybrid cloud, the more agile the response to regulatory requirements. A *business constraint* is a situation that affects the profitability of an enterprise and therefore makes data processing compliance to business policy highly desirable in hybrid cloud

[1] https://resources.flexera.com/web/media/documents/rightscale-2019-state-of-the-cloud-report-from-flexera.pdf

Table 1: Partial trace involving case IDs, activity, timestamp and the processing location (route) during cloud bursting.

#	Case ID	Activity	Timestamp	Route	Activate
LE1	PatientA	consult	Jun 12, 2021 8:25	EDC	
LE2	PatientA	lab	Jun 12, 2021 8:30	EDC	
LE3	PatientA	diagS	Jun 12, 2021 8:30	EDC	con(idc)
LE4	PatientA	consult	Jun 12, 2021 8:35	IDC	
LE5	PatientB	consult	Jun 12, 2021 8:35	EDC	
LE6	PatientB	lab	Jun 12, 2021 8:40	EDC	
LE7	PatientB	diagN	Jun 12, 2021 8:40	EDC	
LE8	PatientB	consult	Jun 12, 2021 8:45	EDC	
LE9	PatientA	pharm	Jun 12, 2021 8:50	IDC	con(bill_idc)
LE10	PatientB	pharm	Jun 12, 2021 8:50	EDC	
LE11	PatientA	bill	Jun 12, 2021 9:15	EDC	$viol_{con(idc)}$
LE12	PatientC	consult	Jun 12, 2021 9:15	EDC	
LE13	...	...	...	...	...

adoption. For example, a business constraint can be applied to a hybrid cloud-based hospital information system to process strictly in the private portion of the hybrid cloud (i.e., in the internal data center) data deemed to be sensitive. Sensitive data in this case can be determined as patient bills that have a specific sequence of activities such as a set of laboratory results preceding a final diagnosis. As another example, business constraints can be applied in the domain of open banking where regulatory requirements enjoin traditional banking organizations to monitor third-party provider services for compliance. In both cases, programming dynamic and complex constraints ahead of time will be unreasonable to implement due to the inherently procedural nature of algorithms, making them unsuitable for detecting constraint-violating events in a hybrid cloud.

In Chesani et al. [8], a hybrid cloud task scheduling algorithm dynamically determines violations to business constraints through process mining of event data. Previously in our own work [9], we present a framework utilizing the Hopcroft-Karp algorithm and Event Calculus (a logic-based framework for process mining) to schedule tasks in a hybrid cloud with consideration to their sensitivity. In another previous work [10] of ours, we present a data- and process-aware framework for scheduling tasks in the hybrid cloud. These studies consider the sensitivity of the tasks to decide the processing location for the tasks in the hybrid cloud infrastructure, firstly to meet deadlines and secondly to comply with given business constraints. In this paper, we investigate the impact of the application of process-aware framework on the the performance of the task scheduling (load-balancing) system.

Table 1 illustrates a partial trace of event data generated from a hybrid cloud-based hospital information system. The constraint *con(idc)* defined as the set of rules: (1) Whenever there is a sensitive diagnosis, subsequent activities involving the case id are to be processed strictly in the internal data center (IDC). (2) All "bill" activities in the IDC are to be processed at least T_0 minutes and not more than T_1 minutes after the "pharm" activity. We define

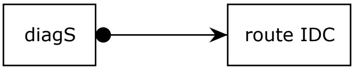


Fig. 1: Declare notation of business constraint indicating that activity "Route IDC" should always follow activity "diagS".

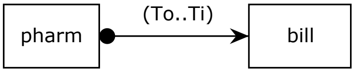


Fig. 2: Declare notation of business constraint indicating that activity "bill" should occur at least T_0 minutes and at most T_1 minutes after the occurrence of activity "pharm".

the constraint using the Declare [11] notation in Figure 1 and Figure 2. Strictly processing all such sensitive tasks in the IDC has the potential for violation of the constraints especially in the cases when the capacity of the resource is limited and unable to accommodate the workload within the required time.

To the best of our knowledge, few modeling studies tackle task scheduling in the hybrid cloud with respect to dynamic business rules. Alcaraz-Mejia et al. [12] and Sourvalas et al. [13] employ Colored Petri Nets (CPNs) in their model for task and resource scheduling, respectively. Their work presents a compact model that can be simulated but does not consider the sensitivity of tasks or dynamic business constraints.

A preliminary version of this paper appears as [14], where we present task scheduling experiments based on the Openstack Octavia load-balancer. However, in our preliminary work we did not consider multiple business constraints. This paper proposes a CPN-based model for task-scheduling in a hybrid cloud with respect for compliance to dynamic business constraints, using a process mining approach. We validate the CPN-based model with a simulation both within the CPN and as an experimental setup. The main contributions of this paper can be summed up as follows:

1. Design and implementation of a CPN model for simulating load-balancing in hybrid clouds under the influence of process mining.

2. Investigation into impact of the proportion of sensitive tasks and number of virtual machines on the performance of a process-mining influenced load-balancer.

3. A systematic analysis of processing time per percentage tasks sensitivity to determine optimal levels of private cloud capacity that supports business constraint compliance.

4. Application of a framework for scale-up capacity of software systems to validate a CPN simulation model for a process mining-influenced load-balancer.

The rest of the paper provides related works in Section 2; Section 3 introduces the background to CPN, process mining, and load balancing; Section 4 presents our approach to the CPN model showing CPN derivation from the Octavia load-balancer system. Results from our simulations and real-world testing are presented and analyzed in Sections 5 and 6, respectively. We present an applicative scenario and conclude the paper in Section 7.

2 Related Works

This section introduces selected studies on workload distribution and load balancing in the hybrid cloud with business constraints of cost, performance, security, or privacy. The presentation classifies the attributes of the works based on: (1) model and approach employed, (2) whether the objective was cost- or efficiency-driven, (3) its implementation type, (4) whether it is data (task) sensitive, and (5) their ability to respond to static and dynamic business constraints. Table 2 captures the distribution of the works according to the classifications. From our reviews, the problem of scheduling tasks in the hybrid cloud with location and data sensitivity has been addressed employing varying approaches including map-reduce, load balancing, machine learning, and also by using process mining middleware. Pluggable simulation frameworks such as CPNs have also been used to facilitate the experimentation of various scheduling configurations of the hybrid cloud.

Works [8, 17–20] employ the MapReduce framework to optimize task scheduling in a hybrid cloud setup. Oktay et al. [18] present a sensitivity model for scheduling where the user is required to mark the sensitive tasks for the algorithm to distinguish between them. In both [17] and [19], their provisioning is dynamic with constraints fixed. Chesani et al. [8] focus on monitoring MapReduce applications in the hybrid cloud with a logic-based process mining tool. Taking inspiration from Business Process Management (BPM) techniques [28], they specify constraints for their application to dynamically scale up the infrastructure to meet QoS requirements. Balagoni and Rao [20] focus on the time- and cost-optimal location to process workloads in a hybrid cloud but do not mention workload sensitivity.

Bittencourt et al. [15] and Abrishami et al. [16] describe DAG-influenced heuristic scheduling algorithms that keep sensitive tasks in the private cloud but the sensitivity of the task is not automatically determined by the algorithms. Statistical and artificial intelligence methods are employed by [21] and [22] to aid task scheduling algorithms. The studies, however, do not discriminate between the sensitivity of tasks being processed.

Load-balancing as a task scheduling mechanism is utilized by [23] and [24]. Sharma et al. [23] present a load-balancing algorithm that mimics the way bats (representing workloads) locate and identify their prey (representing VMs) but does not take into account workload sensitivity during routing. Gandhi et al. [24] present *YODA*, a load-balancing-as-a-service (LBaaS) designed to

Table 2: Our work positioned with respect to the related works focusing on hybrid cloud task scheduling

Paper	approach/ model	cost/ efficiency objective	middle-ware/ tool	data/ task sensi-tivity	static (business) const-raints	dynamic (business) const -raints
Bittencourt (2012) [15]	DAGs	✓				
Abrishami (2015) [16]	DAGs	✓		✓		
Mattess (2013) [17]	MapReduce, Provisioning Policy	✓				
Oktay (2015) [18]	MapReduce, Dynamic Partitioning	✓		✓		
Mao (2016) [19]	MapReduce, Max-Min Strategy	✓				
Balagoni (2016) [20]	MapReduce, Independent Tasks	✓				
Chesani (2017) [8]	MapReduce, Process Mining	✓	✓		✓	✓
Zhang (2014) [21]	FastTopK Algorithm, Popularity pre-filtering	✓				
Ghobaei-Arani (2018) [22]	Reinforcement Learning	✓	✓	✓		
Sharma (2018) [23]	Load-Balancing, Bat Algorithm	✓				
Gandhi (2016) [24]	Load-Balancing, Layer 7, Virtual IPs	✓	✓		✓	
Aktas (2018) [25]	Monitoring metrics, Logging	✓				
Liu & Li (2015) [26]	Stratified Monitoring	✓				
Alcaraz-Mejia (2018) [12]	Binary Decision Tree, CPN	✓				
Souravlas (2018) [13]	Simulation, CPN	✓				
Silva (2020) [27]	Simulation, CPN	✓	✓			
Our Previous Work (2020) [14]	Process Mining, CloudSim	✓	✓	✓		
This article	Process Mining, Load-Balancing, CPN	✓	✓	✓		✓

overcome the limitation of existing layer 7 (L7) load-balancers. The simulations from the study produced four times more redundancy from the application of LBaaS, but the paper, to the best of our understanding, does not include the effect of user policies on the performance of the service.

In utilizing Colored Petri Net as middleware or simulation tool, [12, 13, 27, 29] model task scheduling and resource allocation in hybrid and multi-clouds. The works focus on maximizing utilization and performance of the cloud resources but do not discriminate on the type or sensitivity of tasks during scheduling.

Studies that propose a monitoring framework approach to task scheduling include [9, 14, 25, 26]. Liu and Li utilize agent technology but present no experiments on the implementation and evaluation of the model and its efficiency. In our previous works [9, 14], we present a process mining monitoring mechanism that can influence scheduling in the hybrid cloud towards achieving a more desirable and proportionate VM spawning. Aktas [25] proposes a monitoring software architecture and its capacity to handle large volumes of events, but the monitoring rules are preset. Awada [30] presented a hybrid cloud federation approach that focuses on packing application onto a resource in order to maximize its utilization, rather than allowing the default scheduler to only award resources based on their availability. Federation involves the allocation of resources through a common standard on a contractual basis (SLA) where a provider deems their internal capacity has reached its limit.

The foregoing review show a gradual shift from procedural programming of algorithms to the utilization of data science or statistical techniques for monitoring and controlling task scheduling in the hybrid cloud. Starting with the DAG, the shortest and longest paths aided in scheduling critical resources in data processing systems. The MapReduce-based approach to scheduling in the hybrid cloud introduced efficiency into processing large data sets in parallel but the mechanism largely lacked data-awareness in the studies surveyed. Using artificial intelligence, the approach sought to introduce monitoring mechanisms and efficiency into task scheduling algorithms but appears to lack a generally accepted framework for the dynamic control of algorithms. Load-balancing as a task scheduling mechanism has the potential to route efficiently between portions of the hybrid cloud. Incorporating process mining based monitoring mechanisms can facilitate data- and process-awareness for enhanced scheduling. However, the process of varying hybrid cloud configurations becomes expensive in the real world, at least for organizations looking to establish the proportion of internal data center capacity that will serve their current business constraints. Simulation tools such as the CPN enable such configurations to be determined inexpensively. Table 2 shows the spread of related works done in task scheduling and resource provisioning in a multi- or heterogeneous cloud situation.

```
May 31 18:48:14 10.0.0.1:37318 0/0/4/3/16 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientA"
May 31 18:48:15 10.0.0.1:37320 0/0/3/2/16 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientB"
May 31 18:48:16 10.0.0.1:37322 0/0/14/18/48 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1 caseid:patientA"
May 31 18:48:18 10.0.0.1:37324 0/0/3/2/8 200 40 - - ---- 0/0/0/0/0 0/0 "GET /consult HTTP/1.1 caseid:patientC"
May 31 18:48:19 10.0.0.1:37326 0/0/3/4/18 200 40 - - ---- 0/0/0/0/0 0/0 "GET / HTTP/1.1 caseid:patientA"
May 31 18:48:20 10.0.0.1:37328 0/0/3/2/18 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1 caseid:patientB"
May 31 18:48:21 10.0.0.1:37330 0/0/8/6/14 200 40 - - ---- 0/0/0/0/0 0/0 "GET / HTTP/1.1"
May 31 18:49:15 10.0.0.1:37344 0/0/5/1/6 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1"
May 31 18:49:16 10.0.0.1:37350 0/0/4/2/8 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1"
May 31 18:49:17 10.0.0.1:37352 0/0/4/2/19 200 40 - - ---- 0/0/0/0/0 0/0 "GET /lab HTTP/1.1"
```

Fig. 3: Incoming raw Log event data before being captured by the Logstash log transform component.

```
patientA, /consult, May 31 18:48:14, 10.0.0.1
patientB, /consult, May 31 18:48:15, 10.0.0.1
patientA, /lab,     May 31 18:48:16,  10.0.0.1
patientC, /consult, May 31 18:48:18, 10.0.0.1
patientA, /,        May 31 18:48:19, 10.0.0.1
patientB, /lab,     May 31 18:48:20, 10.0.0.1
```

Fig. 4: Output of the transformation of the log event data as performed by the Logstash log transform component.

3 Background

This section presents an overview of modeling a load-balancer using Colored Petri Nets and influencing the sensitivity of the model with process mining. We first describe the load-balancer modeled after the OpenStack Octavia framework and then relate how we apply process mining to influence its routing policies. The section concludes with an introduction of the CPN model of the load-balancer.

3.1 The Load-Balancer with Process Mining

In cloud computing, load balancing effectively utilizes flexible algorithms with influencing factors such as cost, latency, security, privacy, and compliance. In this paper, we model Octavia[2], a Load Balancing as a Service (LBaaS) project in OpenStack[3]. The Octavia project features *HAProxy*, which is influenced by layer seven (L7) policies and rules to provide "intelligence" at the application level. The log output of HAProxy, on which we leverage to trigger L7 policies, contains a minimum of four attributes — `case id`, `activity`, `resource`, and `timestamp` — upon which the business constraint is applied to determine compliance via process mining [31–33]. In our study, we simulate a process mining component, an extension of Mobucon [31], that reads event data logs to determine conformance or otherwise (violations) by checking the sequence of events against specified constraints that define such sequences' proper order. The output of the process miner is encoded as an API call to create a

[2] `docs.openstack.org/octavia/latest/reference/introduction.html`
[3] docs.openstack.org

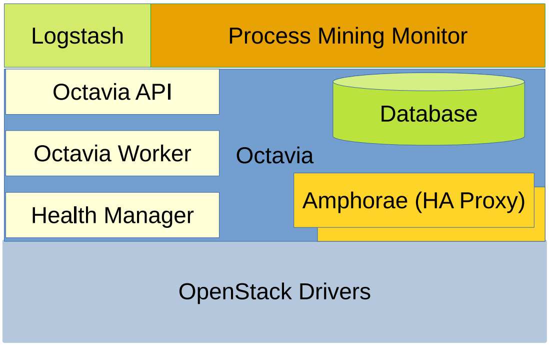


Fig. 5: Architecture of the process mining-influenced load-balancer featuring components of Octavia, an OpenStack Project.

policy or business rule for influencing the load-balancer. We employ *Logstash*, a tool capable of ingesting event data in raw format as shown in Figure 3, and outputting a specified format for feeding into the process mining monitor. Figure 4 shows a typical formatted output of Logstash from the raw input shown in Figure 3. Finally, Figure 5 shows the block diagram of the log transformation tool in relation to the process mining and load-balancer components. The next subsection presents the modeling framework for representing all the individual components.

3.2 Business Constraint Specification

We utilize the "predicates" [31] formalism in Event Calculus [34] and "happenings" [11] on the life cycle of event activities for specifying our business constraints. An instance of a constraint with identification *id*, activity *a* and occurrence at time *t* is generally expressed as *i(id,a,t)* and we feature the status of an activated constraint as either *pending*, *satisfied* or *violated*, denoting the relationship with the event's occurrence. Therefore, *state(i(id,a,t), pend)* is a pending instance of a constraint and the predicate *happens(ev(id,type,a),t)* describes the occurrence of event *ev*, where *type* is the type of event. To illustrate the construction of a constraint we utilize the partial event data from a hospital information systems in Table 1 and its associated set of rules illustrated in Figure 1 and 2. The constraint *con(idc)* is "activated" upon the occurrence of a *diagS* activity:

$$initiates(complete(diagS, CId, Out[Patient, StrictIDC]),\\ status(i(CId, con(idc)), pend), T) \leftarrow StrictIDC = 1$$

and the constraint status changes from the pending state to the satisfied state through:

$$terminates(start(pharm, CId2, Out2, [\ ]), status(i(CId, con(idc)), pending), T)\\ \leftarrow holds\_at(status(i(CId, con(idc)), pend), T) \wedge$$

$happens(complete(diagS, CId, Out, [Patient, Strict]), T_d) \wedge Patient = Out2 \wedge T > T_d \wedge T =< T_d + T_1.$

$initiates(start(pharm, CId2, Out2, [\,]),$
$status(i(CId, con(idc)), satisfied), T) \leftarrow$
$holds\_at(status(i(CId, con(idc)), pending), T) \wedge$
$happens(complete(diagS, CId, Out, [Patient, Strict]), T_d) \wedge Patient = Out2 \wedge$
$T > T_d \wedge T =< T_d + T_1.$

The preceding business constraint *con(idc)* thus monitors at time T for the occurrence of the *diagS* activity, with timestamp T_d and with case id *CId*, and calculates if the activity is happening within the stipulated timeframe $T_d < T < T_d + T_1$. This study is inspired by the Mobucon EC framework [31], which takes as input the event log and outputs the status of the constraint as either satisfied or violated at the end of the life cycle of an event. We provide some more background to the Mobucon Event Calculus specification of business constraints in the supplementary material.

3.3 Colored Petri Nets (CPNs)

To represent the load-balancer as a testable model, we make use of CPNs, which provide avenues for modeling and simulating concurrent workflows in a complete and structured manner [35–37]. We use the primitive structures of the CPN, consisting of *places*, *transitions*, *arcs*, and *colored tokens* to describe the state of the load-balancer and employ the supporting programming language, ML, in assigning and controlling data among the structures. The formal definition of a CPN is provided in the supplementary material.

In load-balancing architectures, scalability is an important goal to ensure high availability for all manner of workload situations [38]. To scale the load-balancer without the need to re-program algorithms is, therefore, a highly desirable feature in the cloud computing paradigm.

4 Proposed Approach to Modeling the Load-Balancer

In the following subsections, we introduce the key elements of the framework within which we construct, simulate, and analyze the performance of our proposed CPN model.

4.1 Framework of the Approach

In the workflow overview provided in Figure 6, each of the main components of the process mining-influenced load-balancer ingests data in one form, processes the data, and outputs the results for ingestion by the next component in the

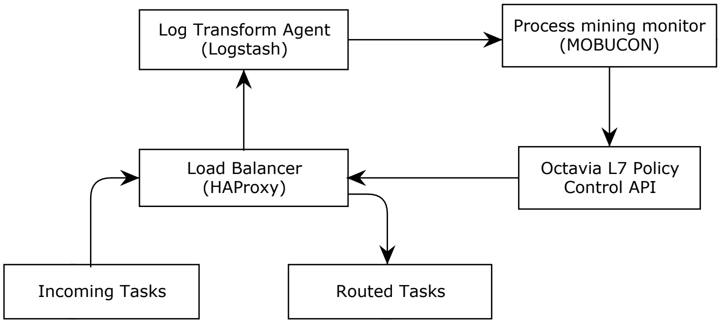


Fig. 6: High-level structure of the workflow components of the load-balancing tool-chain as presented in our previous paper [14], made up of a process mining monitor, the event log agent, the L7 Policy control, and a router.

tool-chain. The result of the cyclical flow of data is in continuous monitoring by the process mining monitor (PMM) for compliance to a specified set of business constraints. From Figure 6, identified violations cause the L7 Policy Controller to create policies to influence the load-balancer to route subsequent tasks in order to avoid further violations. In Section 3.1, we have referred to L7 policies in load-balancers that route requests at the OSI reference model's application layer. The policies can impact the scalability of the overall LBaaS framework and the response time (QoS) for processing tasks in the private cloud. Under the simulations of our proposed CPN model, the response times of the processing tasks in the private cloud are measured for varying workloads and private cloud capacity in terms of the number of VMs. We view the tool-chain as an embedded software within the cloud operating system and draw the initial high-level dataflow overview as in Figure 6. Representing our dataflow overview as a Petri net initially yields the high-level structure in Figure 7, which will be described more in details in the next sections.

4.2 Formulation of the Problem

Often, increasing the capacity of a data center in terms of the quantity of VMs culminates in a proportional reduction in latency for load-balancing, especially when the constraints influencing the load-balancer cause routing overwhelmingly into the internal or private data center. We propose that the greater the number of business constraints the greater the number of sensitive tasks that will result and therefore the greater the overall processing time. Thus $T \propto K * N_s/V_{idc}$, where T = overall processing time, K = number of constraints, N_s = number of sensitive tasks, and V_{idc} = number of VMs in the internal datacenter. Moreover, if $N = N_n + N_s$, where N_n = non-sensitive tasks, then for a number of tasks N of $x\%$ sensitivity, we can calculate $N_s = N * x/100$, which leads to $T \propto K * N * x/V_{idc}$.

In investigating the time-responsiveness of our proposition, we apply the

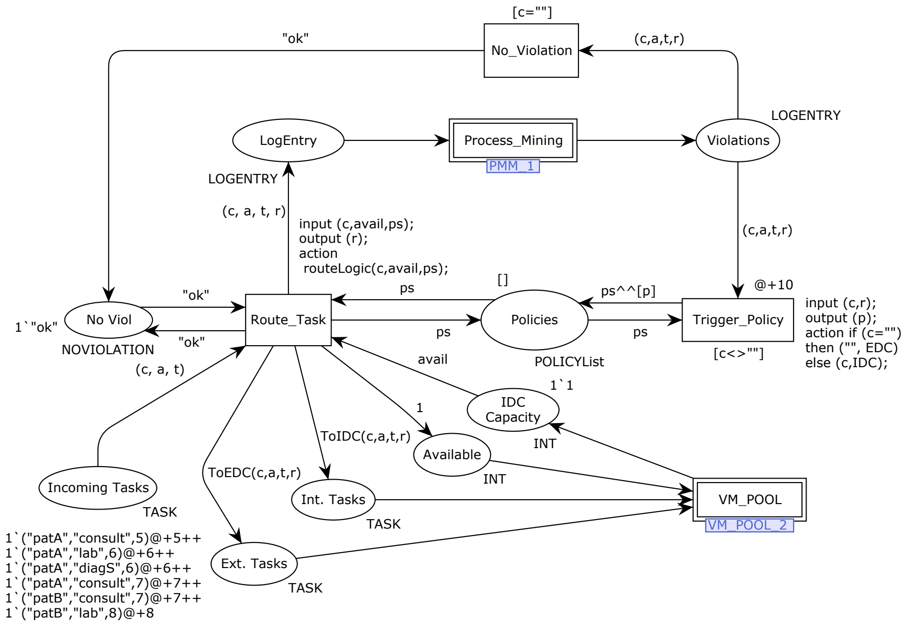


Fig. 7: High-level CPN Model of the process mining-influenced load-balancer comprising routing, process mining and VM Pool components.

universal scalability law from Gunther [38] to ascertain the effect of the changing parameters. In scaling up the LBaaS, the throughput should naturally increase linearly with the increasing of the resource instances. Gunther defines the scale-up capacity C_{sw} of a software system as

$$C_{sw} = \frac{N}{1 - \alpha(N-1) + \beta N(N-1)},$$

where N is the number of tasks to be processed by the software instance; α is a serial fraction, defined as the contention delay for a non-parallel resource; β is known as the coherency parameter, namely the overhead in keeping coherence of the values of a shared resource.

We estimate the parameters α and β by simulation of the CPN and following the procedural steps outlined by Gunther, which also involves setting N to about six selected values, approximately in $\{1, 4, 8, 12, 16, 24, 32, 48\}$ [38]. Plotting the capacity function C_{sw} and interpolating the values shows the response curve of the CPN model to increasing business constraints.

4.3 CPN Model Proposition

We present the full model of our proposed CPN in Figure 7, composed of *transitions* denoting load-balancer system actions; *places* denoting the state of system data flow; *colored tokens* representing the system data types; *arcs* representing the direction of flow of data, and *inscriptions* denoting the conditions under which data can flow between the system components. A detailed description of the initial markings, arcs, and inscriptions for our CPN is provided in the supplementary material.

Transitions Denotation

The main transitions in our high-level CPN model are `Route_Task`, denoting the load-balancer's Amphorae (HAProxy); `Process_Mining`, representing the event data analyzer with business constraint management; `Trigger_Policy`, denoting the Octavia L7 Policy API; and `VM_Pool`, representing the scheduled workload. The `Process_Mining` and `VM_Pool` transitions represented at a high level in Figure 7 have their respective details in Figures 8 and 9.

Places Denotation

The places in the high level CPN are `Incoming Tasks`, denoting the workload awaiting scheduling by the load-balancer; `LogEntry`, representing the transformed event data generated by the load-balancer's Amphorae; `Policies`, representing the L7 policies generated by the Octavia API; `Int.Tasks`, denoting the workload routed to the private cloud; `Ext.Tasks`, representing the workload routed to the public cloud; `Violations`, stores the outcomes of the process mining activity; and `Available`, indicating whether the private cloud capacity is full or otherwise.

Colored Tokens Denotation

The colored tokens in our high level CPN are

- `TASK` having data tuple `(c,a,t)`
- `LOGENTRY` having data tuple `(c,a,t,r)`
- `POLICYList` having the set of data tuples `(c,r)`
- `NOVIOLATION` a count of non-violations,
- `CONSTRAINT` having a preset tuple `(c,a,t,r)`

where the variable `c` denotes the *case id*, `a` denotes the *activity*, `t` denotes the *timestamp* of the activity, `r` denotes the *route* of the workload scheduling, and ε denotes an empty string ("") or absence of a case id specification in a tuple.

Each of the data tuples constitute a *colorset* and is represented by the colored tokens transmitted on the arcs of the `Route_Task` transition.

Table 3: Code Segments of the CPN model's Route Task component

Code Segment for the Route Task Transition

```
fun GetPolicyCaseId (c,_) = c;
fun routeLogic (c,avail,nil) =
if avail = 0 then EDC else IDC
| routeLogic(c,avail,ph::pt) =
if c = GetPolicyCaseId(ph) then IDC
else routeLogic(c,avail,pt);
fun ToIDC(c,a,t,r) =
if r=IDC then 1‘(c,a,t) else empty;
fun ToEDC(c,a,t,r) =
if r=EDC then 1‘(c,a,t) else empty;
```

Code Segment for the Trigger Policy Transition

```
fun triggerPolicy(c, r)=
if |c| = 0
then (ε, EDC)
else (c,IDC)
```

The Route Task Transition

The routing decision on `Route_Task` transition is influenced by the availability of tokens at the `Incoming Tasks`, `IDC Capacity`, `No Viol`, and `Policies` places. The tokens from the `No Viol` and `Policies` places are replaced immediately after the execution. The token `avail` from `IDC Capacity` takes on value `0` or `1` indicating whether `Queue` place capacity is reached or otherwise. The output on the ($Route\_Task \rightarrow Available$) arc is always `1` to indicate the router is ready for the next execution and also to ‘activate’ the `Check_Cap` transition. Section 4.3 describes the `Check_Cap` transition in detail. The `TASK` token is passed on to `Int.Tasks` or `Ext.Tasks` based on the `routeLogic` function which calculates the scheduled route as ‘IDC’ or ‘EDC’ from

$$r = \begin{cases} EDC, & \text{if } avail = 0 \wedge c \notin GetPolicyCaseId(ps) \\ IDC, & \text{if } avail = 0 \wedge c \in GetPolicyCaseId(ps) \\ IDC, & \text{if } avail = 1, \end{cases}$$

where:

$$c = \begin{cases} \varepsilon, & \text{if } |c| = 0 \\ caseid & \text{if } |c| > 0. \end{cases}$$

Table 3 provides the code segments for the definition of the `GetPolicyCaseId(ps)` function. The `POLICYList` tokens are generated by the `Trigger_Policy` tran-

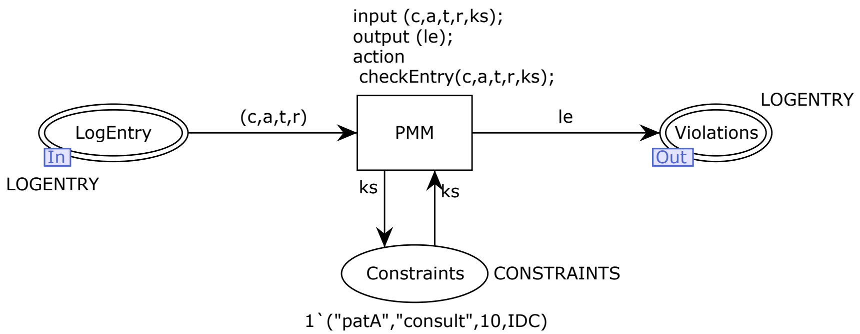


Fig. 8: Details of the Process mining monitor component of the CPN Model depicting a specified business constraint.

sition. The transition function computes a policy `(c,r)` as

$$(c, r) = \begin{cases} (\varepsilon, EDC), & \text{if } |c| = 0 \\ (c, IDC), & \text{otherwise.} \end{cases}$$

The guard function $c <> \varepsilon$ further reinforces the filtering on the transition by allowing only log entries having 'non-empty' case ids. A complete definition for the `Route_Task` transition is provided in the supplementary material.

The Process Mining Monitor (PMM) Transition

The `PMM` transition has two main inputs: `LOGENTRY` tokens consisting of tuple `(c,a,t,r)` and `CONSTRAINTS` tokens denoted by the variable `ks`. The `CONSTRAINTS` tokens are preset at the start of the simulation, providing an overall percentage sensitivity of the incoming tasks. An entry in the constraint list such as `"patA","consult",10, IDC` indicates that any task with case id "patA" processing an activity "consult" on or after ten-time units should be forwarded to the IDC VM Pool. This means that if any task with case id "patA" and timestamp earlier than ten is routed to the EDC VM Pool, then there is no violation.

Figure 8 shows how the list of constraints is immediately put back after the execution of the transition. This ensures the list remains intact and in the same order. The combination of the log entry and constraint list facilitates the execution of the process mining function `checkEntry`. The details of the code segment controlling the `PMM` transition are provided in the supplementary material.

The VM Pool Transition

`VM Pool` represents the back-end servers that process requests forwarded by the router. The VMs in the pool are located in both the private and public

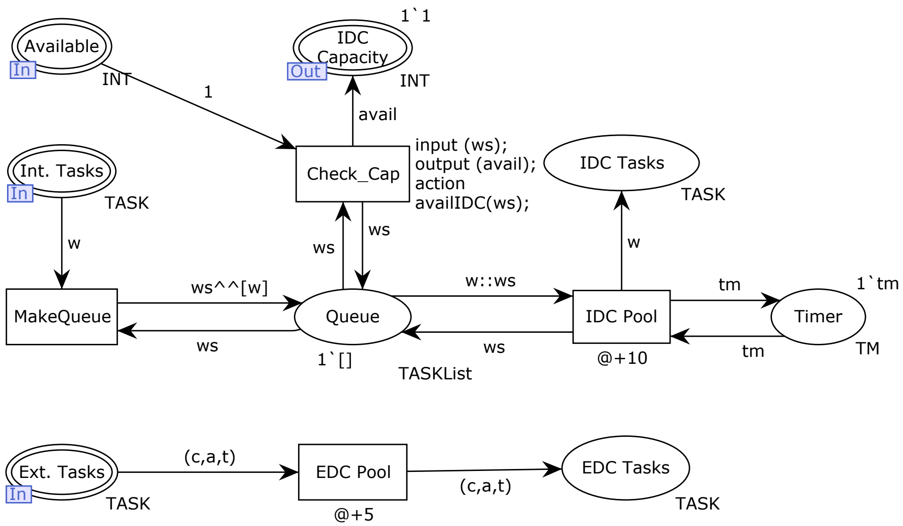


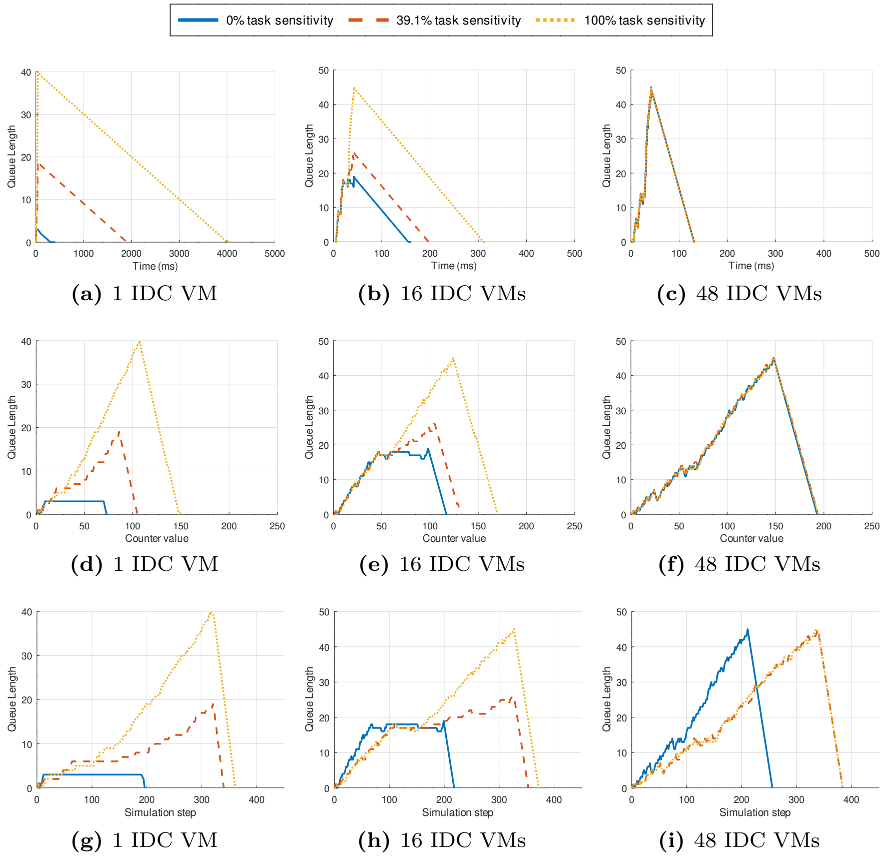
Fig. 9: Details of the VM Pool component of the CPN Model depicting an internal resource-constrained private datacenter (IDC) and an external public cloud (EDC).

portions of the hybrid cloud. The VM Pools in the private cloud are limited in capacity.

From Figure 9, a combination of transitions sends the processed tasks either to the `Int. Tasks` or `Ext. Tasks` place. The `Check_Cap` transition monitors the availability of slots in the queue by comparing the queue length to the number of VMs, and outputs the `avail` variable as `1` if queue length is less and `0` otherwise. The `IDC Pool` transition delays the execution by `D` time units. In the specific example shown in Figure 9, we see that $D = 10$. The `Timer` server is updated with this delay, which signals the next time the `IDC Pool` can transmit the next `TASK` token. The `EDC Pool` transition's delay is five-time units representing an assumed bandwidth limitation in accessing the public cloud. A complete definition of the `VM Pool` transition is provided in the supplementary material.

5 Results

A total of 64 simulations were ran with varying percentage number of sensitive tasks and number of IDC VMs. The number of IDC VMs was set through a variable parameter taking on the values 1, 2, 4, 8, 12, 16, 24, 32 and 48. In each simulation, the CPN model had initial markings of 64 tasks and with the constraints set between zero and six to obtain 0%, 20.3%, 39.1%, 64.1%, 73.4%, 90.6% and 100% as percentages of sensitive tasks. A simple process mining algorithm took as input the set of business constraints and determined



(a) 1 IDC VM **(b)** 16 IDC VMs **(c)** 48 IDC VMs

(d) 1 IDC VM **(e)** 16 IDC VMs **(f)** 48 IDC VMs

(g) 1 IDC VM **(h)** 16 IDC VMs **(i)** 48 IDC VMs

Fig. 10: Plots for task sensitivities of 0%, 39.1% and 100% showing the rate at which the queue length changes per time with (a) 1 IDC VM, (b) 16 IDC VMs and (c) 48 IDC VMs; per counter value with 1, 16 and 48 IDC VMs - (d),(e) and (f); and simulation step when the number of IDC VMs is 1, 16, and 48 - (g), (h), and (i), respectively.

whether an incoming task was sensitive or otherwise. The simulations sought to measure the count of the tasks processed, the number of steps for each task processed, and the time units taken to process each task to the final place in the IDC or EDC VM. The queue length at the IDC VM pool was also observed to measure the queue length per time, the simulation count and the simulation step.

In our results, the simulation counter relates to a place that has a set of tokens, simulation step refers to the number of steps at which data values were observed, and simulation time represents the model's time at which the data values were observed.

5.1 Queue Length per Time

We present results of the token count in the `Queue` place to measure the queue length per time in the IDC VM Pool. Figures 10a, 10b, and 10c show the queue length peaking at 3, 19, and 44 for a VM Pool of 1, 16, and 48, respectively, all at 0% task sensitivity. At 39.1% task sensitivity the queue reaches a peak of 19, 25, and 44 with 1, 16, and 48 IDC VMs, respectively. There is an average peak of 43 `TASK` tokens when task sensitivity is at 100%, irrespective of the number of IDC VMs. The rate of processing the `TASK` tokens is $40/4000 = 0.01$ tasks per millisecond, with 1 IDC VM for all levels of task sensitivity. For the 16 and 48 IDC VMs configuration, the rate of processing the `TASK` tokens is approximately $44/(310 - 44) = 0.165$ and $44/(135 - 45) = 0.489$ tasks per millisecond, respectively, for all levels of task sensitivity.

5.2 Queue Length per Simulation Counter Value

Figures 10d, 10e, and 10f show the queue length per count value of the `Queue` place. The queue lengths peak at 4, 19, and 45 for a VM Pool of 1, 16, and 48, respectively, at 0% task sensitivity. For the task sensitivity of 39.1%, the queue length peaks at 18, 25, and 43 with 1, 16, and 48 IDC VMs, respectively. The queue length peaks per simulation counter value measured at 100% sensitivity are in the range 40 – 44 for all IDC VM configurations. The queue stays full in approximately 65 counts for 1 and 16 IDC VMs at 0% task sensitivity, as it can be seen in Figure 10d and 10e. The rate of change of queue length per counter value is the same for all task sensitivity levels in the 48 IDC VMs.

5.3 Queue Length in 16 IDC VM Configuration

When the number of IDC VMs is set to 16, processing at 0%, 39.1%, and 100% task sensitivity accommodates more tasks in the queue, peaking at 18, 26, and 45 tokens, respectively, as presented in Figure 10b, 10e, and 10h. The queue length rises and peaks less sharply compared with the 1 IDC VM configuration. The processing is also faster in the 16 IDC VMS compared with the 1 IDC VM configuration, completing the simulation at 155ms, 200ms, and 310ms at 0%, 39.1%, and 100% task sensitivity, respectively.

5.4 Queue Length in 48 IDC VM Configuration

For the 48 IDC VM configuration, the queue length peaks at 45 at the 0%, 39.1%, and 100% task sensitivity, as depicted in Figure 10c and 10f. This implies the queue capacity is enough to absorb any percentage of task sensitivity for the number of tasks used in the simulation. In Figure 10i, the queue is processed fastest at the 0% task sensitivity for 48 IDC VMs. This trend applies also to other IDC VM configurations, with the 100% task sensitivity processed slowest across all IDC VM configurations. This indicates the extra simulation steps employed to handle sensitive tasks as shown in Figures 10g, 10h, and 10i.

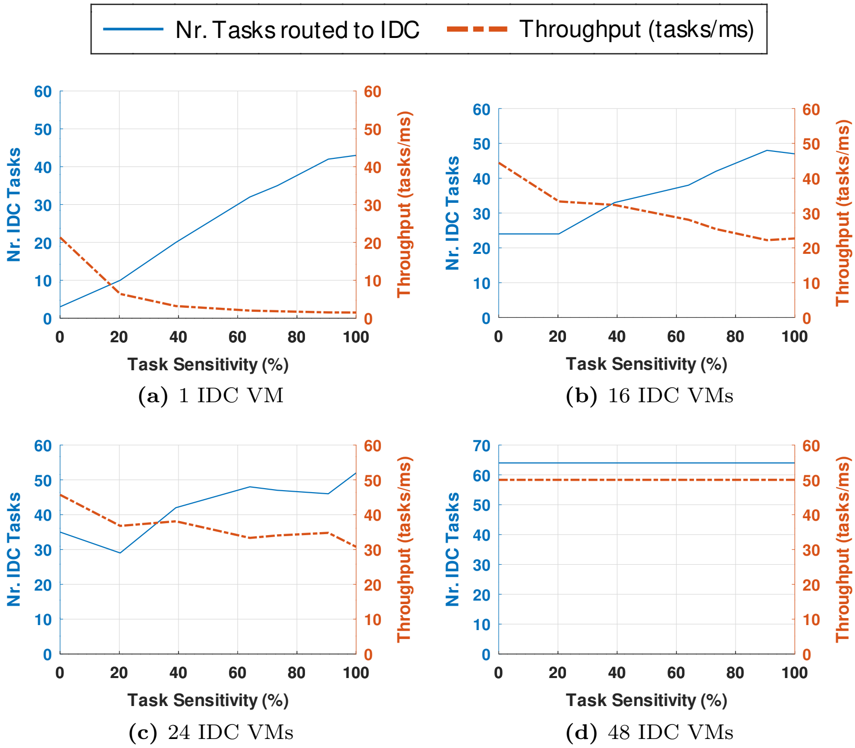


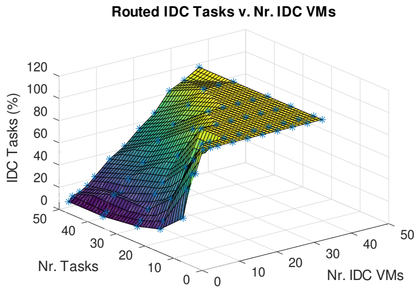
Fig. 11: Throughput and number of TASK tokens routed to the IDC Pool per percentage task sensitivity for (a) 1, (b) 16, (c) 24, and (d) 48 IDC VM configurations, respectively.

5.5 Throughput and Distribution of the `TASK` Tokens

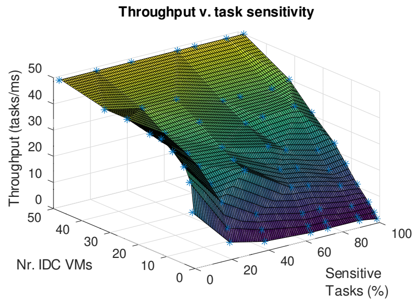
Figure 11 shows the throughput and distribution of `TASK` tokens sent to IDC VM Pool per percentage task sensitivity. Generally, throughput tends to fall as the percentage task sensitivity rises for all IDC VM configurations except that of the 48 IDC VM-configuration, Figure 11d. More `TASK` tokens are also sent to the IDC VM Pool and less to the EDC VM Pool as the percentage task sensitivity increases. Among the trends, the 48 IDC VM-configuration show a non-changing level of distribution of `TASK` tokens per time, 50 tasks/ms, for increasing percentage sensitivity of tasks.

5.6 Change in Processing Time, Throughput, and Cost per Percentage Sensitivity of Tasks and Number of IDC VMs

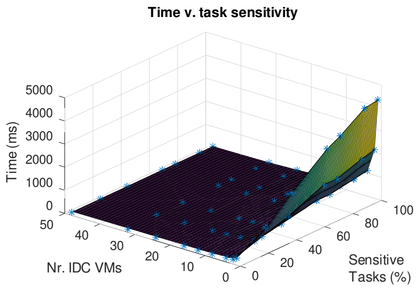
Figure 12a shows that the percentage of `TASK` tokens rises to the maximum as the total number of `TASK` tokens decrease and the number of IDC VMs increase.



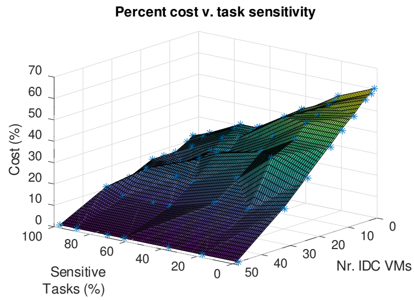
(a) Percentage of TASK tokens sent to the IDC VM Pool vs. number tasks and number of IDC VMS.



(b) Throughput of TASK tokens to both the IDC and EDC VM Pools vs. percentage of sensitive tasks and number of IDC VMS.



(c) Processing time of TASK tokens to both the IDC and EDC VM Pools vs. percentage of sensitive tasks and number of IDC VMS.



(d) Percentage of hybrid cloud cost attributable to rental of public cloud resources vs. percentage sensitivity of tasks and number of IDC VMs.

Fig. 12: Plots on effects of percentage task sensitivity on (a) overall processing time of tasks, (b) throughput of `TASK` tokens routed to both the IDC and EDC VM Pools, (c) percentage of `TASK` tokens routed to the IDC VM Pools and (d) the percentage of the overall operating cost attributable to `TASK` routing to the EDC VM Pool.

This reflects the maxing out of the queue capacity of the IDC VM Pool as the number of IDC VMs is changing.

Figure 12b shows that beyond the 20 IDC VMs mark, the throughput, which is the overall number of `TASK` tokens sent to both the IDC and EDC VM Pools per time, gradually increases but remains not significantly impacted by the change in the percentage sensitive tasks beyond 20%. However, there is a significant rate of change in throughput below 20% sensitive tasks and when the approximate number of IDC VMs is less than 38. This is attributable to the `TASK` tokens sent to the EDC VM Pool on account of low percentage sensitivity and small number of IDC VMs (queue capacity).

In Figure 12c, the plot shows that the workload processing time largely remains constant from the 20 IDC VMs mark for all the percentages of task

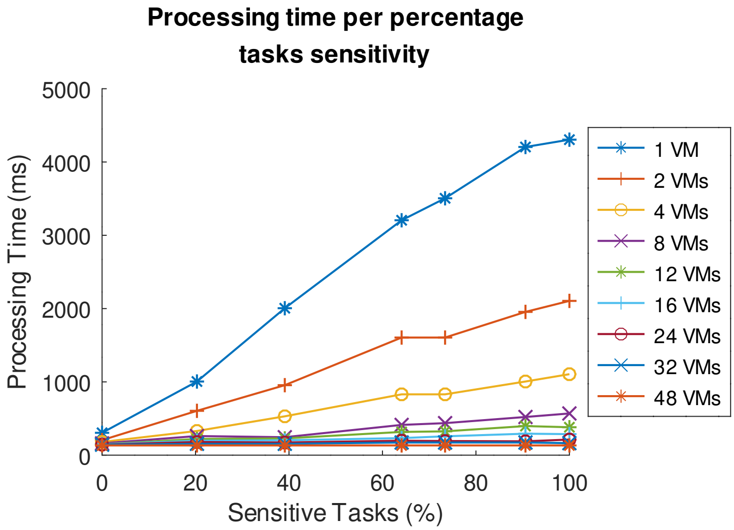


Fig. 13: Comparison of processing times for various numbers of IDC VMs per percentage of sensitive tasks.

sensitivity. Below the number 20 IDC VMs, the processing time is significantly impacted as the percentage of sensitive tasks increases.

Figure 12d displays the characteristic of when there are fewer IDC VMs and fewer sensitive tasks, the system prefers to process the tasks in the EDC VM pool since the capacity of the queue in the IDC VM is maxed out. The percentage of cost attributable to the EDC task processing therefore generally rises as the number of IDC VMs decrease.

Figure 13 shows a finer graduation of the number of IDC VMs per processing time per percentage sensitive tasks. For the 1, 2, and 4 VMs configuration, there is a significant increase in the processing time as the percentage of sensitive tasks increases. The increase in processing times for the 8 VMs and above configuration are marginal, changing only slightly, especially for 40% and below sensitive tasks.

6 Performance Evaluation

The performance of the model is measured in terms of the state (distribution of colored tokens) of the model per time. The main goal of our model is to measure the response of the `Route_Task` transition given the level of the `TASKList`, `POLICYList`, and `avail` tokens. The first response measurement verifies the correct routing of tokens to `IDC Tasks` or `EDC Tasks` place. The second measurement verifies the correct creation of the `POLICY` token based on violation detection. The third measurement verifies the correct creation of the `avail` token from the IDC VM Pool. Finally, the scalability of the model is measured from the overall processing time with respect to the number of business constraints.

6.1 Router Performance

The impact of `POLICYList` set of tokens on the execution of the `Route_Task` transition is depicted in Figures 11a, 11b, and 11c. It shows a decreasing routing to the EDC VM Pool with increasing sensitivity of incoming tasks, indicating that the more the business constraints the more `Route_Task` transition is forced to send `TASK` tokens to `Int.Tasks` place, thereby increasing the overall processing time. This processing time characteristic is shown in Figure 13. From the figure, an administrator of the hybrid cloud would be able to determine or predict the number of IDC VMs needed to accommodate an overall processing time for a given number of tasks. Thus, $\textit{percent tasks} \propto \textit{time taken}$ for any fixed number of VMs, N_{VM}. Also, for any fixed percentage of sensitive tasks, $\textit{time taken} \propto 1/N_{VM}$.

6.2 Policy Trigger Performance

The one-to-one relationship between the number of violating case ids and the number `POLICY` tokens generated by the `Trigger_Policy` transition verifies the latter's correct functioning. Each additional policy potentially increases the overall sensitivity of tasks to be executed. `Trigger_Policy` appends `POLICYList` and gives a clue of number of violations processed, and this complements the number of `No Viols` tokens.

6.3 The Process Mining Monitor Performance

The `PMM` transition acts as a analyser for each output produced by the `Route_Task` via the `LogEntry` place. The larger the number of `CONSTRAINT` tokens, the greater the number of simulation steps to compute the `PMM` transition's output token. This serial processing, as opposed to the parallel execution done at the router, is necessary because the entire log has to be considered to make a decision about whether a log entry violates the constraints. The extra simulation steps taken in sending the tokens to the `Violations` place is evident in Figures 10g to 10i.

6.4 VM Pool Performance

The `IDC Pool` transition takes its inputs from the `Queue` place and is enabled as long as there is a token this place. Our model parallelizes the IDC VM Pool operations by varying the time delay in executing a token. The increasing delay causes a queue length to increase because of slower execution. The model sets the number of VMs N_{VM} via the delay D variable as $N_{VM}\ \alpha\ 1/D$ and this is evident in the queue length increasing more rapidly in smaller number of VMs and also more prolonged processing after queue reaches its peak, as presented in Figure 10.

From the performance of the components, the `POLICYList` truly represents the number of constraints applied on the load-balancer's routing algorithms and the `TASKList` effectively represents the workload on the load-balancer. The higher the quantities and these parameters, the more time needed generally to route data to their final destination.

In summary, the CPN model obeys the scalability law where parallelization of the model as a whole experiences limitations on account of serial aspects of the process mining monitoring. Thus, the more the business constraints, the lower the capacity for scalability of the process mining influenced load-balancer. This is evident in the extra simulation steps taken to process constraint tokens (Figures 10g to 10i).

7 Conclusion and Future Work

This study has modeled and simulated a process mining-influenced load balancer in a hybrid cloud setup. The simulations have investigated the response of the load balancer under increasing proportions of task sensitivity and increasing number of VMs in the private portion of the hybrid cloud. The main response performance indicator observed is the throughput, measured as the number of processed tasks per time under varying increasing number of business constraints. The simulations revealed a direct relationship between the number of business constraints and the percentage of sensitive tasks; and the higher the proportion of tasks that are sensitive, the more private cloud VMs are required to process incoming tasks in order to satisfy the business constraints and SLAs.

One avid adopter of the hybrid cloud is the hospital setting that needs to process its bill periodically. The patient record is tagged as sensitive or otherwise, to determine its processing location within the hybrid cloud. To meet QoS requirements of response time, bill processing is done in the public cloud when the capacity of the on-premises data center is maxed out. The hospital in a time of epidemic may need more data center capacity to maintain QoS requirements. Our simulation setup, based on the process mining of event data, helps to successfully determine the level of resources to engage to meet QoS requirement and comply with regulatory requirement of data privacy.

From the CPN model specified, hybrid cloud adopters have a visual tool with a robust mathematical foundation that can aid in planning and optimizing private cloud resources under frequently changing business data processing constraints. Linking the CPN model to collect and analyze data via OpenStack Octavia is being considered for future work.

References

[1] J. Weinman, "Hybrid Cloud Economics," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 18–22, jan 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7420473/

[2] K. K. Azumah, L. T. Sorensen, and R. Tadayoni, "Hybrid Cloud Service Selection Strategies: A Qualitative Meta-Analysis," in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*. IEEE, aug 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8506887/

[3] B. Wang, C. Wang, Y. Song, J. Cao, X. Cui, and L. Zhang, "A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds," *Cluster Computing*, vol. 23, no. 4, pp. 2809–2834, dec 2020.

[4] C. Baun and M. Kunze, "A Taxonomy Study on Cloud Computing Systems and Technologies," in *Cloud Computing*, ser. Cloud Computing. CRC Press, oct 2011, pp. 73–90.

[5] N. Chopra and S. Singh, "Deadline and cost based workflow scheduling in hybrid cloud," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, ser. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 840–846.

[6] S. U. Khan and N. Ullah, "Challenges in the adoption of hybrid cloud: an exploratory study using systematic literature review," *The Journal of Engineering*, vol. 2016, no. 5, pp. 107–118, may 2016.

[7] G. A. Gravvanis, J. P. Morrison, D. Petcu, T. Lynn, and C. K. Filelis-Papadopoulos, "Special Issue: Recent trends in cloud computing," pp. 700–702, feb 2018.

[8] F. Chesani, A. Ciampolini, D. Loreti, and P. Mello, "Map reduce autoscaling over the cloud with process mining monitoring," in *Communications in Computer and Information Science*, vol. 740. Springer, Cham, apr 2017, pp. 109–130. [Online]. Available: http://link.springer.com/10.1007/978-3-319-62594-2{_}6

[9] K. K. Azumah, S. Kosta, and L. T. Sørensen, "Scheduling in the hybrid cloud constrained by process mining," in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, ser. Proceedings of the International Conference on Cloud Computing Technology and Science (CloudCom). IEEE, dec 2018, pp. 308–313.

[10] K. K. Azumah, L. T. Sørensen, R. Montella, and S. Kosta, "Process mining-constrained scheduling in the hybrid cloud," *Concurrency and*

Computation: Practice and Experience, vol. 33, no. 4, p. e6025, feb 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/cpe.6025

[11] M. Pesic, H. Schonenberg, and W. M. van der Aalst, "DECLARE: Full Support for Loosely-Structured Processes," in *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, ser. 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007). IEEE, oct 2007, pp. 287–287.

[12] M. Alcaraz-Mejia, R. Campos-Rodriguez, and M. Caballero-Gutierrez, "Modeling and Simulation of Task Allocation with Colored Petri Nets," in *Computer Simulation*, D. Cvetkovic, Ed. InTech, jun 2017, ch. 10, pp. 213 – 253. [Online]. Available: http://dx.doi.org/10.5772/67950

[13] S. Souravlas, S. Katsavounis, and S. Anastasiadou, "On Modeling and Simulation of Resource Allocation Policies in Cloud Computing Using Colored Petri Nets," *Applied Sciences*, vol. 10, no. 16, p. 5644, aug 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/16/5644

[14] K. K. Azumah, S. Kosta, and L. T. Sørensen, "Load balancing in hybrid clouds through process mining monitoring," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11874 LNCS. Springer, nov 2019, pp. 148–157.

[15] L. F. Bittencourt, E. R. Madeira, and N. L. Da Fonseca, "Scheduling in hybrid clouds," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 42–47, 2012.

[16] H. Abrishami, A. Rezaeian, G. K. Tousi, and M. Naghibzadeh, "Scheduling in hybrid cloud to maintain data privacy," in *Fifth International Conference on the Innovative Computing Technology (INTECH 2015)*, 2015, pp. 83–88.

[17] M. Mattess, R. N. Calheiros, and R. Buyya, "Scaling MapReduce Applications Across Hybrid Clouds to Meet Soft Deadlines," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, ser. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE, mar 2013, pp. 629–636. [Online]. Available: http://ieeexplore.ieee.org/document/6531813/

[18] K. Y. Oktay, S. Mehrotra, V. Khadilkar, and M. Kantarcioglu, "SEMROD: Secure and Efficient MapReduce Over HybriD Clouds," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '15. ACM, 2015, pp. 153–166. [Online]. Available: http://doi.acm.org/10.1145/2723372.2723741

[19] X. Mao, C. Li, W. Yan, and S. Du, “Optimal Scheduling Algorithm of MapReduce Tasks Based on QoS in the Hybrid Cloud,” in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, 2016, pp. 119–124.

[20] Y. Balagoni and R. R. Rao, “A cost-effective SLA-aware scheduling for hybrid cloud environment,” in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2016, pp. 1–7.

[21] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen, “Proactive workload management in hybrid cloud computing,” *IEEE Transactions on Network and Service Management*, vol. 11, no. 1, pp. 90–100, mar 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6701292/

[22] M. Ghobaei-Arani, S. Jabbehdari, and M. A. Pourmina, “An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach,” *Future Generation Computer Systems*, vol. 78, pp. 191–210, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.02.022

[23] S. Sharma, S. Verma, K. Jyoti, and Kavita, “Hybrid bat algorithm for balancing load in cloud computing,” *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4.12 Special Issue 12, pp. 26–29, oct 2018. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/20986

[24] R. Gandhi, Y. C. Hu, and M. Zhang, “Yoda: A highly available layer-7 load balancer,” in *Proceedings of the 11th European Conference on Computer Systems, EuroSys 2016*. New York, New York, USA: ACM Press, 2016, pp. 1–16. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2901318.2901352

[25] M. S. Aktas, “Hybrid cloud computing monitoring software architecture,” *Concurrency and Computation*, vol. 30, no. 21, p. e4694, nov 2018. [Online]. Available: http://doi.wiley.com/10.1002/cpe.4694

[26] Y. C. Liu and C. L. Li, “A Stratified Monitoring Model for Hybrid Cloud,” *Applied Mechanics and Materials*, vol. 719-720, no. Materials and Engineering Technology, pp. 900–906, jan 2015. [Online]. Available: https://www.scientific.net/AMM.719-720.900

[27] F. A. Silva, I. Fé, and G. Gonçalves, “Stochastic models for performance and cost analysis of a hybrid cloud and fog architecture,” *Journal of Supercomputing*, pp. 1–25, may 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11227-020-03310-1

[28] W. Van der Aalst, J. Desel, and A. Oberweis, *Business Process Management: Models, Techniques, and Empirical Studies.* Springer, 2000, vol. 1806.

[29] F. A. Silva, S. Kosta, M. Rodrigues, D. Oliveira, T. Maciel, A. Mei, and P. Maciel, "Mobile cloud performance evaluation using stochastic models," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1134–1147, 2018.

[30] U. Awada, "Hybrid cloud federation : A case of better cloud resource efficiency," 2018. [Online]. Available: https://www.researchgate.net/profile/Uchechukwu-Awada/publication/324886932_Hybrid_Cloud_Federation_A_Case_of_Better_Cloud_Resource_Efficiency/links/5ae92f5245851588dd8162e2/Hybrid-Cloud-Federation-A-Case-of-Better-Cloud-\Resource-Efficiency.pdf

[31] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. van der Aalst, "Monitoring business constraints with the event calculus," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 1, pp. 1–30, dec 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2542182.2542199

[32] F. M. Maggi, M. Dumas, L. García-Bañuelos, and M. Montali, "Discovering data-aware declarative process models from event logs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8094 LNCS. Springer, Berlin, Heidelberg, 2013, pp. 81–96. [Online]. Available: http://link.springer.com/10.1007/978-3-642-40176-3{_}8

[33] R. De Masellis, F. M. Maggi, M. Montali, and M. Montali, "Monitoring data-aware business constraints with finite state automata," in *Proceedings of the 2014 International Conference on Software and System Process - ICSSP 2014*, ser. the 2014 International Conference. New York, New York, USA: ACM Press, may 2014, pp. 134–143. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2600821.2600835

[34] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, pp. 67–95, mar 1986. [Online]. Available: http://link.springer.com/10.1007/BF03037383

[35] Z. Tari, P. Bertok, and A. Mukherjee, *Verification of communication protocols in web services: Model-checking service compositions.* Wiley-IEEE, 2013. [Online]. Available: https://www.wiley.com/en-gh/Verification+of+Communication+Protocols+in+Web+Services:+Model+Checking+Service+Compositions-p-9780470905395

[36] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, jun 2007. [Online]. Available: https://link.springer.com/article/10.1007/s10009-007-0038-x

[37] K. Jensen and L. M. Kristensen, *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Springer Publishing Company, Incorporated, 2009.

[38] N. J. Gunther, *Guerrilla capacity planning: A tactical approach to planning for highly scalable applications and services.* Springer Berlin Heidelberg, 2007.

SUPPLEMENTARY MATERIAL

Modeling and Simulating a Process Mining-Influenced Load-Balancer for the Hybrid Cloud

1 The Event Calculus Formalization

The formalization of constraints can be facilitated by the Event Calculus [1] (EC) framework and described by a set of propositions culled from domain theory and also from general theory. The generated propositions are referred to as "predicates" [2] describing a state or action of the variables that make up the constraint. Predicates to represent an initial state, activation, occurrence of an event and the deactivation of a constraint have been denoted respectively as *initially*, *initiates*, *happens* and *terminates*. A predicate *holds_at* gives an indication of an event attribute having a specified value. The predicates facilitate a description of the effect of event occurrences on the constraint variables. With each additional event, a "reasoner" calculates the validity of the constraint variables.

Table 1: Specification of the Sensitive Diagnosis Constraint for MOBUCON showing the status moving from pending (pend) to either violated (viol) or to satisfied (sat).

new instance	$initiates(complete(pharm, CId, P_o, [Pat, Mins]),$ $status(i(CId, con(S)), pend), T) \leftarrow Minutes \leq 60.$
pend to sat	$terminates(start(bill, CId_2, P_{o2}, [\,]),$ $status(i(CId, con(S)), pend), T) \leftarrow$ $holds\_at(status(i(CId, con(S)), pend), T) \wedge$ $happens(complete(pharm, CId, P_o, [Pat, Mins]), T_p) \wedge Pat =$ $P_{o2} \wedge T > T_p \wedge T \leq T_p + 60.$ $initiates(start(bill, CId_2, P_{o2}, [\,]),$ $status(i(CId, con(S)), sat), T) \leftarrow$ $holds\_at(status(i(CId, con(S)), pend), T) \wedge$ $happens(complete(pharm, CId, P_o, [Pat, Mins]), T_p) \wedge Pat =$ $P_{o2} \wedge T > T_p \wedge T \leq T_p + 60.$
pend to viol	$terminates(\_, status(i(CId, con(S)), pend), T) \leftarrow$ $holds\_at(status(i(CId, con(S)), pend), T) \wedge$ $happens(complete(pharm, CId, P_o, [Pat, Mins]), T_p) \wedge T >$ $T_p + 60.$ $initiates(\_, status(i(CId, con(S)), viol), T) \leftarrow$ $holds\_at(status(i(CId, con(S)), pend), T) \wedge$ $happens(complete(pharm, CId, P_o, [Pat, Mins]), T_p) \wedge T >$ $T_p + 60.$
pend to viol	$terminates(ex(complete), status(i(CId, con(X)), pend)).$ $initiates(ex(complete), status(i(CId, con(X)), viol), T) \leftarrow$ $holds\_at(status(i(CId, con(X)), pend), T).$

2 Definition of Colored Petri Nets (CPNs)

A nine-tuple $CPN = (\Sigma, P, T, A, V, C, G, E, I)$ formally defines a CPN, where:

- Σ is a finite set of non-empty *color sets*
- P is a finite set of places
- T is a finite set of transitions and
- A is a finite set of directed arcs: $P \cap T = \emptyset$ and $A \subseteq (P \times T \cup T \times P)$
- $V : Type[v] \in \Sigma \ \forall v \in V$ is a finite set of typed variables
- $C : P \rightarrow \Sigma$ is a *color set function* that assigns a color set to each place
- $G : T \rightarrow EXPR_v$ is a *guard function* that assigns a guard to each transition $t \mid Type[G(t)] = Bool$
- $E : A \rightarrow EXPR_v$ is an *arc expression function* that assigns an arc inscription to each arc $a \mid Type[E(a)] = C(p)_{MS}$, p is the place connected to arc a
- $I : P \rightarrow EXPR_\emptyset$ is an *initialization function* that assigns initialization inscription to each place p such that $Type[I(p)] = C(p)_{MS}$

3 Initial Markings

The initial markings show the starting state of the model. In the CPN model for the process mining influenced load-balancer, the `Incoming Tasks`, `No Viol` and `IDC Capacity` places all have initial tokens that activate the `Route_Task` transition. Each transition must have tokens on the incoming arc in order to be activated, execute and pass on its output data. The initial marking at the `Incoming Tasks` place for our simulation is 64 `TASK` tokens.

4 Arcs and Inscriptions Denotation

The set of out-going arcs of the `Route_Task` transition are $(Route\_Task \rightarrow Policies)$, $(Route\_Task \rightarrow Int.Tasks)$, $(Route\_Task \rightarrow Ext.Tasks)$, $(Route\_Task \rightarrow LogEntry)$, $(Route\_Task \rightarrow Available)$, and $(Route\_Task \rightarrow No\,Viol)$. Incoming arcs are $(Incoming\ Tasks \rightarrow Route\_Task)$, $(No\,Viol \rightarrow Route\_Task)$, $(IDC\ Capacity \rightarrow Route\_Task)$, $(No\,Viol \rightarrow Route\_Task)$, $(Policies \rightarrow Route\_Task)$, $(No\,Viol \rightarrow Route\_Task)$, and $(IDC\ Capacity \rightarrow Route\_Task)$. Both incoming and outgoing arcs to the `Route_Task` transition are listed in Table 2. Each arc is inscribed with expression functions $E(i)$ as detailed out in Table 2. Arc expressions characterize the restrictions or filters imposed on its colorset transmission.

5 The Route Task Transition

Table 2: The CPN model definition for the Route Task component

$$CPN = \{\Sigma, P, T, A, V, C, G, E, I\}$$

$$P = \{LogEntry, NoViol, Policies, IncomingTasks, Ext.Tasks, Int.Tasks, Available, IDCCapacity, Violations\}$$

$$T = \{Route\_Task, Trigger\_Policy, No\_Violation\}$$

$$A = \{(NoViol \rightarrow Route\_Task), (Route\_Task \rightarrow NoViol), (Inc.Tasks \rightarrow Route\_Task), (Route\_Task \rightarrow Int.Tasks), (Route\_Task \rightarrow Ext.Tasks), (Route\_Task \rightarrow Available), (Route\_Task \rightarrow Policies), (Policies \rightarrow Route\_Task), (Trigger\_Policy \rightarrow Policies), (Policies \rightarrow Trigger\_Policy), (Route\_Task \rightarrow LogEntry), (IDCCapacity \rightarrow Route\_Task), (Violations \rightarrow Trigger\_Policy), (Violations \rightarrow No\_Violation), (No\_Violation \rightarrow NoViol)\}$$

$$\Sigma = \{LOGENTRY, NOVIOLATION, TASK, INT, POLICYList\}$$

$$V = \{ps, p, avail, c, a, t, r\}$$

$$C(p) = \begin{cases} TASK, & \text{if } p \in \{IncomingTasks, Ext.Tasks, Int.Tasks\} \\ LOGENTRY, & \text{if } p \in \{LogEntry, Violations\} \\ POLICYList, & \text{if } p \in \{Policies\} \\ NOVIOLATION, & \text{if } p \in \{NoViol\} \\ INT, & \text{if } p \in \{Available, IDCCapacity\} \end{cases}$$

$$G(i) = \begin{cases} routeLogic(), & \text{if } i \in \{Route\_Task\} \\ c = \varepsilon, & \text{if } i \in \{No\_Violation\} \\ c <> \varepsilon, & \text{if } i \in \{Trigger\_Policy\} \\ true, & \text{otherwise} \end{cases}$$

$$E(i) = \begin{cases} (c, a, t), & \text{if } i = (IncomingTasks \rightarrow Route\_Task) \\ (c, a, t, r), & \text{if } i \in \{(Route\_Task \rightarrow LogEntry), (Violations \rightarrow Trigger\_Policy), (Violations \rightarrow No\_Violation)\} \\ ps, & \text{if } i \in \{(Route\_Task \rightarrow Policies), (Policies \rightarrow Route\_Task), (Policies \rightarrow Trigger\_Policy)\} \\ ps\hat{}\hat{}[p], & \text{if } i = (Trigger\_Policy \rightarrow Policies) \\ "ok", & \text{if } i \in \{(No\_Violation \rightarrow NoViol), (NoViol \rightarrow Route\_Task), (Route\_Task \rightarrow NoViol)\} \\ ToEDC() & \text{if } i = (Route\_Task \rightarrow Ext.Tasks) \\ ToIDC() & \text{if } i = (Route\_Task \rightarrow Int.Tasks) \\ 1 & \text{if } i = (Route\_Task \rightarrow Available) \\ avail & \text{if } i = (IDCCapacity \rightarrow Route\_Task) \\ \varnothing, & \text{otherwise} \end{cases}$$

The CPN model definition for the Route Task component (continued from Table 2)

$$I(p) = \begin{cases} 1`(''patA'', ''con'', 5)@+5++ & \\ 1`(''patA'', ''lab'', 6)@+6++ & \\ .... & \text{if } p \in \{IncomingTasks\} \\ 1`''ok'' & \text{if } p \in \{NoViol\} \\ 1`1 & \text{if } p \in \{IDCCapacity\} \\ 1`[] & \text{if } p \in \{Policies\} \\ \varnothing, & \text{otherwise} \end{cases}$$

Code Segment for the Route Task Transition

```
fun GetPolicyCaseId (c,_) = c;
fun routeLogic (c,avail,nil) =
if avail = 0 then EDC else IDC
| routeLogic(c,avail,ph::pt) =
if c = GetPolicyCaseId(ph) then IDC
else routeLogic(c,avail,pt);
fun ToIDC(c,a,t,r) =
if r=IDC then 1`(c,a,t) else empty;
fun ToEDC(c,a,t,r) =
if r=EDC then 1`(c,a,t) else empty;
```

Code Segment for the Trigger Policy Transition

```
fun triggerPolicy(c, r)=
if |c| = 0
then (ε, EDC)
else (c,IDC)
```

6 The VM Pool Transition

Table 3: The CPN Model definition for the VM Pool component

$$CPN = \{\Sigma, P, T, A, V, C, G, E, I\}$$

$$P = \{Queue, Timer, IDCTasks, EDCTasks, Available, IDCCapacity, Int.Tasks, Ext.Tasks\}$$

$$T = \{Check\_Cap, MakeQueue, IDCPool, EDCPool\}$$

$$\begin{aligned} A = \{&(Int.Tasks \rightarrow MakeQueue), (MakeQueue \rightarrow Queue), \\ &(Queue \rightarrow IDCPool), (Queue \rightarrow MakeQueue), \\ &(IDCPool \rightarrow Queue), (Timer \rightarrow IDCPool), \\ &(IDCPool \rightarrow IDCTasks), (Ext.Tasks \rightarrow EDCPool), \\ &(EDCPool \rightarrow EDCTasks), (Available \rightarrow Check\_Cap), \\ &(Check\_Cap \rightarrow Queue), (Check\_Cap) \rightarrow IDCCapacity), \\ &(IDCPool \rightarrow Timer), (Queue \rightarrow Check\_Cap)\} \end{aligned}$$

$$\Sigma = \{TASK, TASKList, INT, TM\}$$

$$V = \{w, ws, tm, avail, c, a, t\}$$

$$C(p) = \begin{cases} LOGENTRY, & \text{if } p \in \{LogEntry, Violations\} \\ CONSTRAINTS, & \text{if } p \in \{Constraints\} \end{cases}$$

$$G(i) = \begin{cases} availIDC(), & \text{if } i \in \{Check\_Cap\} \\ true, & \text{otherwise} \end{cases}$$

$$E(i) = \begin{cases} w, & \text{if } i \in \{(Int.Tasks \rightarrow MakeQueue), \\ & \quad (IDCPool \rightarrow IDCTasks)\} \\ ws, & \text{if } i \in \{(Queue \rightarrow MakeQueue), (IDCPool \rightarrow Queue), \\ & \quad (Check\_Cap \rightarrow Queue), (Queue \rightarrow Check\_Cap)\} \\ ws\hat{}\hat{}[w], & \text{if } i = (MakeQueue \rightarrow Queue) \\ w :: ws, & \text{if } i = (Queue \rightarrow IDCPool) \\ (c, a, t), & \text{if } i \in \{(Ext.Tasks \rightarrow EDCPool), \\ & \quad (EDCPool \rightarrow EDCTasks)\} \\ tm & \text{if } i = (IDCPool \rightarrow Timer) \vee (Timer \rightarrow IDCPool) \\ avail & \text{if } i = (Check\_Cap) \rightarrow IDCCapacity) \\ 1 & \text{if } i = (Available \rightarrow Check\_Cap) \\ \varnothing, & \text{otherwise} \end{cases}$$

$$I(p) = \begin{cases} 1`1 & \text{if } p \in \{IDCCapacity\} \\ 1`tm & \text{if } p \in \{Timer\} \\ 1`[] & \text{if } p \in \{Queue\} \\ \varnothing, & \text{otherwise} \end{cases}$$

Code Segment for the VM Pool Check Capacity Transition

```
fun availIDC(ws) =
if (length ws <= Nvm) then 1 else 0;
```

7 The Process Mining Monitor Transition

Table 4: The CPN model definition for Process mining monitor component.

$$CPN = \{\Sigma, P, T, A, V, C, G, E, I\}$$
$$P = \{LogEntry, Constraints, Violations\}$$
$$T = \{PMM\}$$
$$A = \{(LogEntry \rightarrow PMM), (PMM \rightarrow Violations), (Constraints \rightarrow PMM), (PMM \rightarrow Constraints)\}$$
$$\Sigma = \{LOGENTRY, CONSTRAINTS\}$$
$$V = \{le, ks, c, a, t, r\}$$

$$C(p) = \begin{cases} LOGENTRY, & \text{if } p \in \{LogEntry, Viol\} \\ CONSTRAINTS, & \text{if } p \in \{Constraints\} \end{cases}$$

$$G(i) = \begin{cases} checkEntry(), & \text{if } i \in \{PMM\} \\ true, & \text{otherwise} \end{cases}$$

$$E(a) = \begin{cases} le, & \text{if } a = (PMM \rightarrow Violations) \\ ks, & \text{if } a \in \{(PMM \rightarrow Constraints), (Constraints \rightarrow PMM)\} \\ \varnothing, & \text{otherwise} \end{cases}$$

$$I(p) = \begin{cases} 1`(\text{“}patA\text{”}, \text{“}consult\text{”}, 10, IDC)..., \\ \text{if } p \in \{Constraints\} \\ \varnothing, & \text{otherwise} \end{cases}$$

Code Segment for the Process Mining Transition

```
fun GetCaseId(c, _, _, _) = c;
fun GetActivity(_, a, _, _) = a;
fun GetTime(_, _, t, _) = t;
fun GetRoute(_, _, _, r) = r;
fun checkEntry(c, a, t, r, nil) = (ε, a, t, r)
| checkEntry(c, a, t, r, kh :: kt) =
if c = GetCaseId(kh)
and also a = GetActivity(kh)
and also t > GetTime(kh)
and also r <> GetRoute(kh) then (c, a, t, r)
else checkEntry(c, a, t, r, kt);
```

8 Experiment and Simulation Data

Table 5: Constraints applied in the simulation experiment involving 64 tasks

1‘("patA","consult",5,IDC)++
1‘("patB","consult",7,IDC)++
1‘("patC","consult",15,IDC)++
1‘("patD","consult",17,IDC)++
1‘("patE","consult",26,IDC)++
1‘("patF","consult",29,IDC)

Table 6: Data used in the simulation experiment involving 64 tasks (1)

1‘("patA","consult",5)@+5++
1‘("patA","lab",6)@+6++
1‘("patA","diagS",6)@+6++
1‘("patA","consult",7)@+7++
1‘("patB","consult",7)@+7++
1‘("patB","lab",8)@+8++
1‘("patB","diagN",8)@+8++
1‘("patB","consult",9)@+9++
1‘("patA","pharm",10)@+10++
1‘("patB","pharm",10)@+10++
1‘("patA","bill",15)@+15++
1‘("patC","consult",15)@+15++
1‘("patB","bill",16)@+16++
1‘("patC","lab",16)@+16++
1‘("patC","diagS",16)@+16++
1‘("patC","consult",17)@+17++
1‘("patD","consult",17)@+17++
1‘("patD","lab",18)@+18++
1‘("patD","diagN",18)@+18++
1‘("patD","consult",19)@+19++
1‘("patC","pharm",20)@+20++
1‘("patD","pharm",20)@+20++
1‘("patC","bill",25)@+25++
1‘("patC","bill",26)@+26++
1‘("patE","consult",26)@+26++
1‘("patE","lab",28)@+28++
1‘("patE","diagS",29)@+29++
1‘("patE","consult",30)@+30++
1‘("patB","consult",29)@+29++
1‘("patB","lab",31)@+31++

Table 7: Data used in the simulation experiment involving 64 tasks (Continued)

1‘("patB","diagN",32)@+32++
1‘("patD","consult",30)@+30++
1‘("patE","pharm",31)@+31++
1‘("patB","consult",32)@+32++
1‘("patB","pharm",33)@+33++
1‘("patE","bill",32)@+32++
1‘("patC","consult",33)@+33++
1‘("patB","bill",34)@+34++
1‘("patE","consult",29)@+29++
1‘("patF","consult",29)@+29++
1‘("patF","lab",31)@+31++
1‘("patF","diagN",32)@+32++
1‘("patC","consult",30)@+30++
1‘("patE","lab",31)@+31++
1‘("patF","consult",32)@+32++
1‘("patF","pharm",33)@+33++
1‘("patE","consult",32)@+32++
1‘("patC","lab",33)@+33++
1‘("patF","bill",34)@+34++
1‘("patE","pharm",35)@+35++
1‘("patE","bill",35)@+35++
1‘("patC","lab",36)@+36++
1‘("patC","diagS",36)@+36++
1‘("patC","consult",37)@+37++
1‘("patA","consult",37)@+37++
1‘("patA","lab",38)@+38++
1‘("patA","diagN",38)@+38++
1‘("patA","consult",39)@+39++
1‘("patC","pharm",40)@+40++
1‘("patA","pharm",40)@+40++
1‘("patC","bill",41)@+41++
1‘("patC","bill",42)@+42++
1‘("patA","consult",42)@+42++
1‘("patA","bill",42)@+42

References

[1] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, no. 1, pp. 67–95, mar 1986. [Online]. Available: http://link.springer.com/10.1007/BF03037383

[2] M. Montali, F. M. Maggi, F. Chesani, P. Mello, and W. M. P. van der Aalst, "Monitoring business constraints with the event calculus," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 1, pp. 1–30, dec 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2542182.2542199

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-828-1

AALBORG UNIVERSITY PRESS